Inango MDU Management Software

iMMS ver 10.8

User Manual

Inango Systems Ltd.

Feb, 2013

Statement

The information contained in this manual is subject to change without any notice. The company makes no guarantees as to the information referred to in the manual and assumes no responsibility for any errors or damages that may be caused by provision or usage of this document.

The proprietary information contained in the manual is under Copyright Act protection and any rights are reserved by the company. Any reproduction or translation of any part of this manual is strictly prohibited without prior written consent of the company.

Thank you for using products of Inango Systems Ltd.

Relating Icons Description

Several striking icons are used in the manual to indicate aspects requiring special attention. They are used as following:

Cautions of special importance, any ignorance may cause unrecoverable results.

Cautions of importance, should be paid attention to.

Note: information helpful for the understanding of the manual.

Table of Contents

1. Overview **5**
1.1 Main changes from previous version **5**
2. System Architecture **6**
2.1 VLANs **8**
2.2 Management Software **8**
3. General Functions **10**
3.1 Starting with iMMS **10**
3.2 Factory Defaults **11**
3.3 Reset to Defaults / Reboot **16**
3.4 Disaster recovery **17**
3.5 License file **17**
3.6 Software Upgrade **17**
3.6.1 Automatic code upgrades **19**
3.7 Managing Users **20**
3.7.1 Create a New User **20**
3.8 Network Interfaces **21**
3.9 Authentication **21**
3.10 Useful Functions **22**
3.10.1 Topology **22**
3.10.2 Testing Line Quality (Diagnostics) **22**
3.10.3 Cross-Talk **23**
3.10.4 Noise Test **23**
3.10.5 Configuring Rate Limitation **23**
3.10.6 Multi-Cast Operations **23**
3.10.7 Link Aggregation **23**
3.10.8 DHCP Relay **24**
3.10.9 Filters **24**
4. Managed Data **26**
5. Command-Line Interfaces **31**
5.1 Deleting entries in config tables **33**
5.2 Command list **33**
6. HTTP Interfaces **40**
6.1 Overview **40**
6.2 Filters **41**
6.3 Login **41**
6.4 Opening Screen **42**
6.5 System Info Tab **42**
6.5.1 Copy **42**
6.5.2 Generic System Parameters **44**
6.5.3 Auto-upgrade **45**
6.5.4 Topology **46**
6.6 iMMS Info Tab **47**

6.6.1 Interface Table 48
6.6.2 IP Address Table 49
6.6.3 Authentication 50
6.6.4 IMMS VLAN 51
6.6.5 User Management Table 52
6.6.6 iMMS Diagnostics 53
6.6.7 iMMS Settings 55
6.6.8 Time Settings 59
6.6.9 DHCP Relay 60
6.6.10 Routing 61
6.7 Masters Tab 62
6.7.1 Master Table 62
6.7.2 Statistics Table 64
6.7.3 Multicast Info 64
6.7.4 Master Diagnostics 64
6.8 Endpoints Tab 66
6.8.1 Endpoints Table 66
6.8.2 Diagnostics 67
6.8.3 Endpoint Statistic Table 69
6.8.4 Multicast 70
6.8.5 Bandwidth Management 70
6.8.6 Network 71
7. 73
7.1 Copy 73
7.2 Ping 74
7.3 Traceroute 75
7.4 iMMS Configuration Commands 75
7.4.1 IP Configuration 75
7.4.2 VLAN Configuration 76
7.4.3 DHCP Relay 77
7.4.4 Routing Table 78
7.4.5 ARP Table 79
7.5 HPNA Master Commands 79
7.5.1 Configure HPNA Master 79
7.5.2 Multicast Configuration 80
7.6 HPNA Endpoint Commands 80
7.6.1 Configure HPNA Endpoint 80
8. MIB 82

1. Overview

Management is a key part of any product and the Inango MDU Management System (iMMS) is designed to meet your management needs for HPNA-based MDU products. The software may run on a variety of platforms (detailed later) and accessed via either SNMP, HTTP or a Command-Line Interface (CLI).

Moreover, the system is highly extensible, which greatly eases integration of additional functionality like adding managed data or other access protocols such as TR-069, etc.

In the following pages, we explain more of the internals of iMMS and discuss in greater detail the managed entities that are controlled by it on Inango HPNA products.

1.1 Main changes from previous version

The main additions in iMMS Version 10.8 relative to the last official release (10.6) are as follows:

- 802.1x authentication

  The master units can now perform 802.1x authentication. They support Authenticator mode (for authenticating CPE equipment connected to the HPNA endpoints) as well as Supplicant mode (in which the master is authenticated itself)

- Automatic update

  Software (iMMS) and firmware (HPNA) upgrades are now done automatically by depositing the new code versions on a server. IMMS will check for any updates periodically and update all relevant units accordingly

- HPNA tests

  HPNA diagnostic tests for cross talk and noise have been added

2. System Architecture

An HPNA deployment is a layer 2 network: all traffic is directed through the network from the service provider, through an MDU Master Unit (usually located in basements of each building) and MDU Subscriber Endpoints that are located in each apartment. The physical medium used is Ethernet and coaxial cables: the service provider connects to each Master unit via Ethernet (typically over fiber) and the Masters connect to the Endpoints via the coaxial cables that already run throughout the building. Household equipment then connects to the MDU Endpoints via Ethernet but these are already beyond the scope of the HPNA network.

As mentioned, this is a layer-2 network. Data is switched (or bridged) from user devices, through the endpoints and master and up to the service provider. Networking functions such as DHCP Servers and NAT/firewalls usually run at the service provider site (though of course many other options are also possible).

From a data processing viewpoint, there are three main types of entities in the system:

- a management station

  This is the platform on which the managed data is stored and accessed. In iMMS, the management station may be either a computer connected to the HPNA network or (some or all of) the MDU Master units[1].

- MDU Masters

  Each MDU Master is located at some central place in the building (for example, in the basement, though large buildings may have a master on each floor) and connects to multiple endpoints. Bandwidth is shared among all endpoints - you will find many options for rate limitation in the endpoints so that this bandwidth sharing can be controlled[2].

- MDU Endpoint

  An MDU Endpoint bridges data from the coaxial cables to the in-house Ethernet network. The endpoints are registered with each master. The ability to run diagnostics tests on the coaxial cables between the master and endpoints is an important function of the network. These tests are activated through the endpoints via the management software.

[1]Inango offers two types of MDU Masters: managed and non-managed. Managed masters have a processor and include a full software communications stack – and are therefore capable of being a management station in the sense described in this manual. Non-managed masters have no processor and in this case, the management station must be elsewhere (i.e. either a computer or a managed Master connected to the same HPNA network).

[2]Rate limitation at the master units is not currently supported (though may be added in future).

MDU Endpoints typically have two Ethernet ports to which user equipment may be connected.

The following illustration shows how these units may be connected together. In this example, we are assuming that iMMS runs on a managed master (i.e. iMMS runs on the master itself) and that four endpoints have been connected via a splitter.

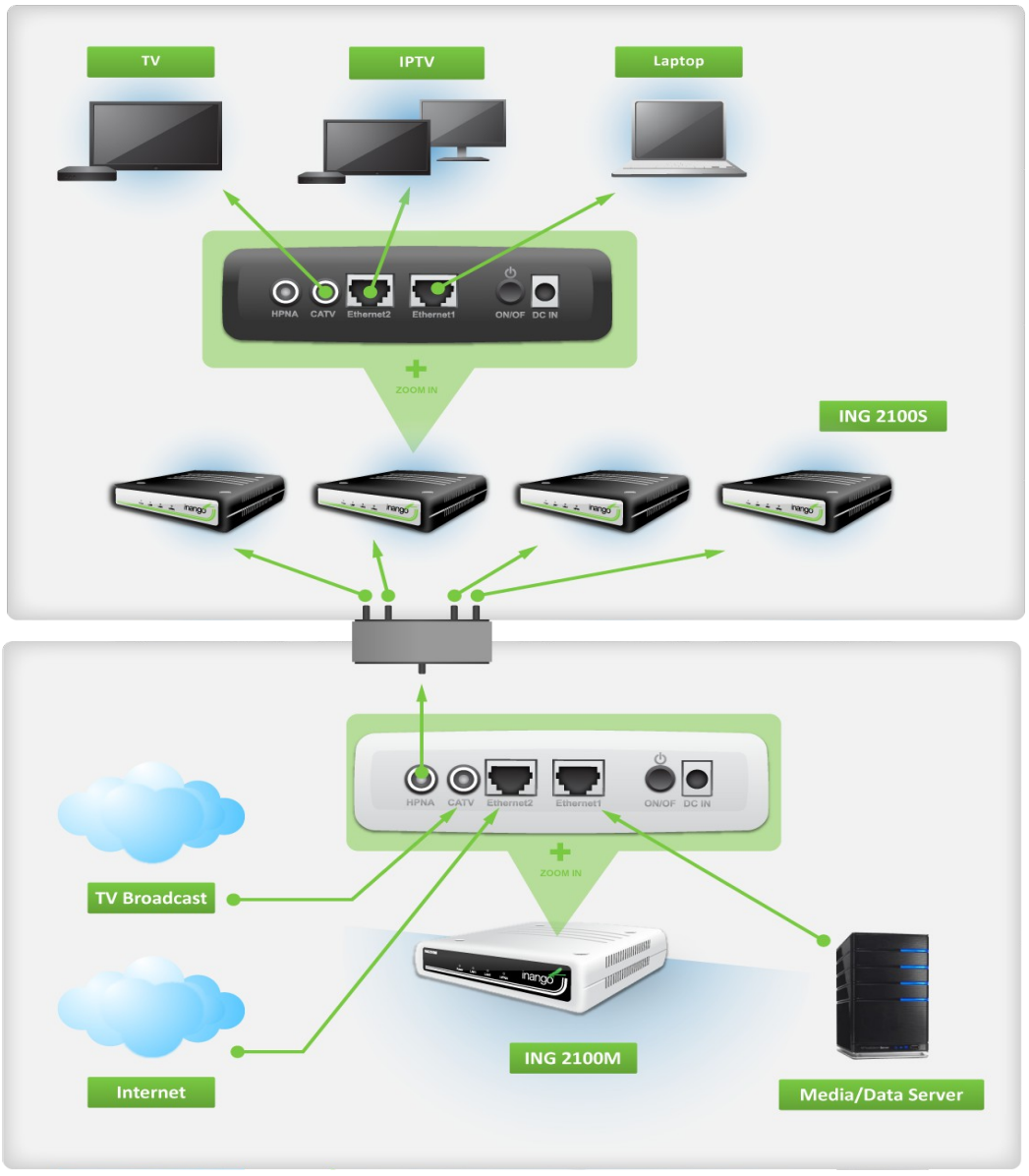


Illustration 1: Connecting HPNA Equipment

2.1 VLANs

Many HPNA deployments use VLANs to separate traffic according to a variety of parameters: VLANs per endpoint, VLAN per Ethernet port on each endpoint, traffic type and so on. VLAN tagging (if done) is done at the endpoints. There are two ways of doing tagging: by using the HPNA controller or by using the Ethernet switch in the endpoint. We do not support doing both simultaneously (i.e. no double tagging). These two tagging functions are configured and activated or deactivated via different managed entities, as detailed in the sections below.

2.2 Management Software

The job of the management software (on whichever platform it runs) is to manage all data in the system. We provide thorough solutions for HPNA-related functions, but the system is highly flexible and extensible – Inango can add more functions for specific systems if/where required.

In general, the managed entities are all stored in an SQL data base. They are accessed via *front-end* engines. We currently support three front-ends, namely SNMP, HTTP and CLI. When using SNMP, run an SNMP manager (or browser) on a PC; then access is via HTTP, the data is accessed via a standard HTTP client (browser) and when using CLI, you need to log in to the management system from a shell (after accessing the unit via either telnet or ssh). In all cases, the same managed data is accessed (in fact you may use several different front-ends simultaneously as all accesses to the managed data are secured).

When data is changed (via any front-end engine), the change may need to be reflected in the underlying system. For example, if you change an IP address on an interface, besides updating an SQL data-base entry, we also need to update the underlying Linux system to use the new IP address (e.g. by running ifconfig). These changes are reflected in the system via *back-end* functions that are associated with each managed data entity. The back-ends are not visible to users and are activated automatically by the system when necessary.

The main elements in our management software system are shown below.

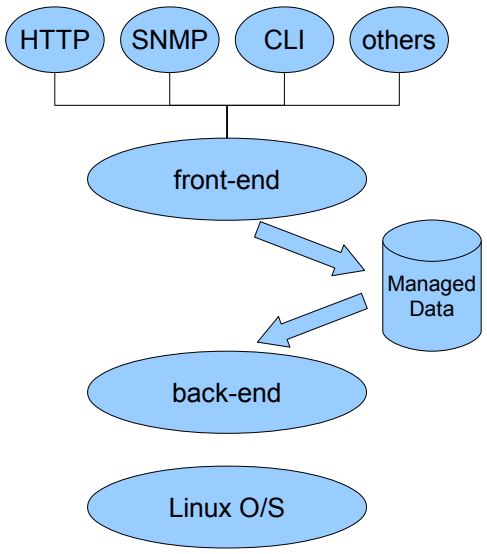


Illustration 2: Management Software Elements

3. General Functions

This section describes the general functions in iMMS. Note that some of these descriptions will vary depending on whether you are connecting to iMMS via SNMP, HTTP or logging in via a CLI.

3.1 Starting with iMMS

IMMS is run automatically when the system on which it is installed is turned on. You then need to access iMMS from some standard mechanism from a computer: either an HTTP Client (web browser), an SNMP Manager (or browser) or by accessing the iMMS system telnet (which logs you in and runs the iMMS CLI interface).

The default IP address of the management system is 192.168.1.2

The information needed to run each of these options is as follows:

HTTP	Enter the address as `http://192.168.1.2` Default username/password are “user” and “changeme” (the default user has administrative rights)
SNMP	Set community to “public” (for get) and to “private” (for set)
CLI	Do “telnet” (or “ssh”) to the IP address of the iMMS station and login with default user/password.

3.2 Factory Defaults

The default IP address of the management system is 192.168.1.2

Default values for all MIB elements are shown in the table below.

ASN.1 Code	Name	Default	Value
1.3.6.1.4.1.9112.1.1.1.1.1.1.1.1	operationId	2975	2975
1.3.6.1.4.1.9112.1.1.1.1.1.1.1.2	sourceProtocol	tftp	3
1.3.6.1.4.1.9112.1.1.1.1.1.1.1.7	destProtocol	local	5
1.3.6.1.4.1.9112.1.1.1.1.1.1.1.14	operationStatus	pending	3
1.3.6.1.4.1.9112.1.1.1.1.1.1.1.15	abort	FALSE	2
1.3.6.1.4.1.9112.1.1.1.1.1.1.1.16	start	TRUE	1
1.3.6.1.4.1.9112.1.1.1.1.1.2.1	systemResetToDefault	noop	2
1.3.6.1.4.1.9112.1.1.1.1.1.2.2	systemReboot	noop	2
1.3.6.1.4.1.9112.1.1.1.1.1.2.3	hpnaSystemResetToDefault	noop	2
1.3.6.1.4.1.9112.1.1.1.1.1.2.4	sysResetToDefaultKeepConn	noop	2
1.3.6.1.4.1.9112.1.1.1.1.1.2.5	sysAutoUpgrade	noop	2
1.3.6.1.4.1.9112.1.1.1.1.1.2.6.1	autoUpgradeTimeInterval	1440	1440
1.3.6.1.4.1.9112.1.1.1.1.1.2.6.2	autoUpgradeTransactionInterval	5	5
1.3.6.1.4.1.9112.1.1.1.1.1.2.6.3	autoUpgradeWebPage	http://www.inango.com/imms/swfw.html	http://www.inango.com/imms/swfw.html
1.3.6.1.4.1.9112.1.1.1.1.1.2.6.4	autoUpgradeEnable	disable	2
1.3.6.1.4.1.9112.1.1.1.1.1.3.3.1.6	epCPETopologyOnlineOnlyInfo	onlineonly	1
1.3.6.1.4.1.9112.1.1.1.1.1.5.1.6	epCustomerTopologyOnlineOnlyInfo	onlineonly	1
1.3.6.1.4.1.9112.1.1.1.1.2.1.1.2	interfaceType	nic	2
1.3.6.1.4.1.9112.1.1.1.1.2.1.1.8	ifaceMngEnable	enable	1

1.3.6.1.4.1.9112.1.1.1.1.2.1.1.9	ifaceHPNAScaningEnable	enable	1
1.3.6.1.4.1.9112.1.1.1.1.2.2.1.4	priority	1	1
1.3.6.1.4.1.9112.1.1.1.1.2.2.1.5	type	dhcp	2
1.3.6.1.4.1.9112.1.1.1.1.2.3.1.1.2	vlanAlias	Vlan Desription	Vlan Desription
1.3.6.1.4.1.9112.1.1.1.1.2.3.2.1.3	vlanPortTagged	untagged	2
1.3.6.1.4.1.9112.1.1.1.1.2.4.1.3	userAccessPermissions	review	1
1.3.6.1.4.1.9112.1.1.1.1.2.5.1.1.2	iMMSDiagnosticControlOperation	ping	1
1.3.6.1.4.1.9112.1.1.1.1.2.5.1.1.3	iMMSDiagnosticControlHostName	www.inango.com	www.inango.com
1.3.6.1.4.1.9112.1.1.1.1.2.5.1.1.5	iMMSDiagnosticControlResult	noop	2
1.3.6.1.4.1.9112.1.1.1.1.2.6.1.1	snmpPublicCommunityString	public	public
1.3.6.1.4.1.9112.1.1.1.1.2.6.1.2	snmpPrivateCommunityString	private	private
1.3.6.1.4.1.9112.1.1.1.1.2.6.1.3.1.3	snmpTrapServersPriority	1	1
1.3.6.1.4.1.9112.1.1.1.1.2.6.1.4	snmpSysLocation	Inango Systems	Inango Systems
1.3.6.1.4.1.9112.1.1.1.1.2.6.1.5	snmpSysContact	contact@inango.com	contact@inango.com
1.3.6.1.4.1.9112.1.1.1.1.2.6.2.1	webLinePerTable	8	8
1.3.6.1.4.1.9112.1.1.1.1.2.6.2.2	webInfoDirection	horizontal	2
1.3.6.1.4.1.9112.1.1.1.1.2.6.4.2	localLogMsgMaxLevel	emergency	0
1.3.6.1.4.1.9112.1.1.1.1.2.6.4.3	localLogMsgMaxNumber	100	100
1.3.6.1.4.1.9112.1.1.1.1.2.6.4.5	localLogMsgEnable	enable	1
1.3.6.1.4.1.9112.1.1.1.1.2.6.4.6	localLogMsgClean	noop	2
1.3.6.1.4.1.9112.1.1.1.1.2.6.4.7.1.3	syslogServerEnable	enable	1
1.3.6.1.4.1.9112.1.1.1.1.2.6.4.8	syslogMsgMaxLevel	emergency	0
1.3.6.1.4.1.9112.1.1.1.1.2.9	iMMSHostname	Inango-iMMS-system	Inango-iMMS-system
1.3.6.1.4.1.9112.1.1.1.1.2.10.1	timeZone	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.10.2	timeFormat	yyyymmddhms	1
1.3.6.1.4.1.9112.1.1.1.1.2.10.4	ntpClientEnable	enable	1
1.3.6.1.4.1.9112.1.1.1.1.2.11	iMMSLicenseInfo	No license	No license

1.3.6.1.4.1.9112.1.1.1.1.2.14	primaryDNSServer	8.8.8.8	8.8.8.8
1.3.6.1.4.1.9112.1.1.1.1.2.15	secondaryDNSServer	8.8.4.4	8.8.4.4
1.3.6.1.4.1.9112.1.1.1.1.2.17.1	iMMSDhcpRelayEnable	disable	2
1.3.6.1.4.1.9112.1.1.1.1.2.17.2.1.3	iMMSDhcpRelaySrvListPriority	1	1
1.3.6.1.4.1.9112.1.1.1.1.2.17.3.1.2	iMMSDhcpRelayMasterEnable	enable	1
1.3.6.1.4.1.9112.1.1.1.1.2.17.3.1.3	iMMSDhcpRelayMasterRemoteIDType	epMac	1
1.3.6.1.4.1.9112.1.1.1.1.2.17.3.1.5	iMMSDhcpRelayMasterCircuitIdType	epMac	1
1.3.6.1.4.1.9112.1.1.1.1.2.19.1.1.3	iMMSRoutingMask	255.255.255.255	255.255.255.255
1.3.6.1.4.1.9112.1.1.1.1.2.19.1.1.6	iMMSRoutingMetric	10	10
1.3.6.1.4.1.9112.1.1.1.1.2.19.1.1.7	iMMSRoutingType	static	1
1.3.6.1.4.1.9112.1.1.1.1.2.19.2.1.5	iMMSArpType	static	1
1.3.6.1.4.1.9112.1.1.1.1.2.20	authenticationGroup	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.1	authenticatorGroup	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.1	eapReauthPeriod	15	15
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.2	eapolVersion	ver2	2
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.3	eapolIpAddr	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.4	nasIdentifier	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.5	primaryAuthServerAddr	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.6	primaryAuthServerPort	1812	1812
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.7	primaryAuthServerSharedSecret	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.8	secondaryAuthServerAddr	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.9	secondaryAuthServerPort	1812	1812
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.10	secondaryAuthServerSharedSecret	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.11	acctServerAddr	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.12	acctServerPort	1813	1813
1.3.6.1.4.1.9112.1.1.1.1.2.20.1.13	acctServerSharedSecret	0	0
1.3.6.1.4.1.9112.1.1.1.1.2.20.2	supplicantGroup	0	0

1.3.6.1.4.1.9112.1.1.1.1.2.20.3	enableAuthenticator	disable	2
1.3.6.1.4.1.9112.1.1.1.1.2.20.4	enableSupplicant	disable	2
1.3.6.1.4.1.9112.1.1.1.1.3.1.1.7	reboot	noop	2
1.3.6.1.4.1.9112.1.1.1.1.3.1.1.8	masterInfoStatus	offline	2
1.3.6.1.4.1.9112.1.1.1.1.3.1.1.10	resetToDefault	noop	2
1.3.6.1.4.1.9112.1.1.1.1.3.1.1.11	resetMasterSubSystemToDefault	noop	2
1.3.6.1.4.1.9112.1.1.1.1.3.1.1.13	masterInfoAuth	authorized	1
1.3.6.1.4.1.9112.1.1.1.1.3.1.1.14	masterLAGEnable	disable	2
1.3.6.1.4.1.9112.1.1.1.1.3.1.1.15	masterInfoFWSig	0	0
1.3.6.1.4.1.9112.1.1.1.1.3.2.1.1.2	masterDiagnosticsRFStartTest	FALSE	2
1.3.6.1.4.1.9112.1.1.1.1.3.2.1.1.3	masterDiagnosticsRFPacketNumberInTest	600	600
1.3.6.1.4.1.9112.1.1.1.1.3.2.1.1.4	masterDiagnosticsRFResetResults	noop	2
1.3.6.1.4.1.9112.1.1.1.1.3.2.1.1.5	masterDiagnosticsRFResetLigthResults	noop	2
1.3.6.1.4.1.9112.1.1.1.1.3.2.2.1.3	masterXTalkTestStatus	non	2
1.3.6.1.4.1.9112.1.1.1.1.3.2.2.1.4	masterXTalkTestStartTest	FALSE	2
1.3.6.1.4.1.9112.1.1.1.1.3.2.2.1.5	masterXTalkTestClearResults	noop	2
1.3.6.1.4.1.9112.1.1.1.1.3.2.2.1.6	masterXTalkTestCrossPackets	0	0
1.3.6.1.4.1.9112.1.1.1.1.3.2.3.1.3	masterNoiseTestHPNAStatus	non	2
1.3.6.1.4.1.9112.1.1.1.1.3.2.3.1.4	masterNoiseTestHPNAStartTest	FALSE	2
1.3.6.1.4.1.9112.1.1.1.1.3.2.3.1.5	masterNoiseTestHPNACleanResults	noop	2
1.3.6.1.4.1.9112.1.1.1.1.3.2.3.1.6	masterNoiseTestHPNANumOfIterations	4	4
1.3.6.1.4.1.9112.1.1.1.1.3.3.1.1.63	masterStatisticRFResetCounters	noop	2
1.3.6.1.4.1.9112.1.1.1.1.3.3.3	resetAllMasterStatistics	noop	2
1.3.6.1.4.1.9112.1.1.1.1.3.3.4.1.2	masterEPDeviceEPSeen	0	0
1.3.6.1.4.1.9112.1.1.1.1.3.3.4.1.3	masterEPDeviceEPOnline	0	0
1.3.6.1.4.1.9112.1.1.1.1.3.3.4.1.4	masterEPDeviceEPAuthorized	0	0
1.3.6.1.4.1.9112.1.1.1.1.3.3.4.1.5	masterEPDeviceEPNonAuthorized	0	0

1.3.6.1.4.1.9112.1.1.1.1.3.3.4.1.6	masterEPDeviceHostsOnline	0	0
1.3.6.1.4.1.9112.1.1.1.1.3.4.1.1.3	masterMcastIGMPSnoopingEnable	disable	2
1.3.6.1.4.1.9112.1.1.1.1.3.4.1.1.4	masterMcastIGMPSnoopingTimeout	30	30
1.3.6.1.4.1.9112.1.1.1.1.4.1.1.7	epInfoReboot	noop	2
1.3.6.1.4.1.9112.1.1.1.1.4.1.1.8	epInfoInfoStatus	offline	2
1.3.6.1.4.1.9112.1.1.1.1.4.1.1.10	epInfoResetToDefault	noop	2
1.3.6.1.4.1.9112.1.1.1.1.4.1.1.12	epInfoTrafficEnabled	TRUE	1
1.3.6.1.4.1.9112.1.1.1.1.4.1.1.13	epInfoMaxNumberOfHosts	5	5
1.3.6.1.4.1.9112.1.1.1.1.4.1.1.14	epInfoIgmpSnoopingEnabled	TRUE	1
1.3.6.1.4.1.9112.1.1.1.1.4.1.1.16	epInfoAuth	authorized	1
1.3.6.1.4.1.9112.1.1.1.1.4.1.1.17	epInfoFWSig	0	0
1.3.6.1.4.1.9112.1.1.1.1.4.2.1.1.3	epDiagnosticHPNAStatus	non	2
1.3.6.1.4.1.9112.1.1.1.1.4.2.3.1.4	epLightDiagnosticHPNAStartTest	FALSE	2
1.3.6.1.4.1.9112.1.1.1.1.4.2.3.1.5	epLightDiagnosticHPNAPacketNumberInTest	60	60
1.3.6.1.4.1.9112.1.1.1.1.4.2.4.1.5	epNoiseTestHPNACleanResults	noop	2
1.3.6.1.4.1.9112.1.1.1.1.4.2.4.1.6	epNoiseTestHPNANumOfIterations	4	4
1.3.6.1.4.1.9112.1.1.1.1.4.2.5	cleanAllEPDiagnostics	noop	2
1.3.6.1.4.1.9112.1.1.1.1.4.2.6	cleanAllEPLightDiagnostics	noop	2
1.3.6.1.4.1.9112.1.1.1.1.4.2.7	cleanAllEPNoiseTestResults	noop	2
1.3.6.1.4.1.9112.1.1.1.1.4.3.1.1.63	epStatisticRFResetCounters	noop	2
1.3.6.1.4.1.9112.1.1.1.1.4.3.3	resetAllEPStatistics	noop	2
1.3.6.1.4.1.9112.1.1.1.1.4.4.1.1.3	epMcastIGMPSnoopingEnable	disable	2
1.3.6.1.4.1.9112.1.1.1.1.4.4.1.1.4	epMcastIGMPSnoopingTimeout	30	30
1.3.6.1.4.1.9112.1.1.1.1.4.5.1.1.5	epHPNAUCUpstreamLimit	0	0
1.3.6.1.4.1.9112.1.1.1.1.4.5.1.1.6	epHPNABCUpstreamLimit	50	50
1.3.6.1.4.1.9112.1.1.1.1.4.5.1.1.7	epHPNAMCUpstreamLimit	0	0
1.3.6.1.4.1.9112.1.1.1.1.4.5.1.1.14	epHPNATotalDownstreamLimit	0	0

1.3.6.1.4.1.9112.1.1.1.1.4.5.1.1.16	epHPNATotalDownstreamPeakLimit	0	0
1.3.6.1.4.1.9112.1.1.1.1.4.7.1.1.2	epHPNAMacVlanEnable	disable	2
1.3.6.1.4.1.9112.1.1.1.1.4.7.1.1.6	epHPNAMacVlanOtherTagPassDrop	enable	1
1.3.6.1.4.1.9112.1.1.1.1.4.7.2.1.4	epNetMacPortTaged	untagged	2
1.3.6.1.4.1.9112.1.1.1.1.4.7.2.1.5	epNetMacVlanTagRemove	enable	1

3.3 Reset to Defaults / Reboot

iMMS may be reset to factory defaults or rebooted.

Note: After a reset operation the system will be automatically rebooted.

HTTP To reset all system values to factory defaults, click the relevant button in the SystemInfo section and push the Apply button.

To reboot the entire system, check the reboot iMMS check box

SNMP To reset all system values to factory defaults, set the systemResetToDefault scalar in SystemGroup->GenericParamsGroup to 1 (true) (OID 1.3.6.1.4.1.40354.1.1.1.1.2.1.0)

To reboot whole system, set the systemReboot scalar in SystemGroup->GenericParamsGroup to 1 (true) (OID 1.3.6.1.4.1.40354.1.1.1.1.2.2.0)

CLI Go to

```
hpna system param
```

then to reboot iMMS, type

```
config reboot 1
```

and to reset values to factory defaults, type

```
config reset 1
```

and to HPNA configuration only, type

```
config hpha-reset 1
```

3.4 Disaster recovery

If the system IP address is lost or the system becomes totally unresponsive, you may perform disaster recovery to either display the current configuration or reset to factory defaults.

To enter the disaster recovery state, power the device off/on and start checking (pinging) IP address 192.168.1.2. When the ping starts working, the master will be in disaster recovery state for 15 seconds. In this state, you may enter any of the following commands:

Exit	Exit the disaster recovery state and continue booting the device as regular
Reboot	Perform a power reset
Reset	Reset all values to factory defaults
Show	Display all currently defined configuration values. Note that since the unit will be in the process of a boot these values may not have been applied.
Stop	Stop the boot process. In this case, the disaster recovery period will continue until you enter a Reboot or Exit command. The 15 second timeout will not apply.

3.5 License file

iMMS comes with a license file that defines its permitted use and how the HPNA devices in the system are to be used. All iMMS systems are shipped with a default license that provides essentially unrestricted use of all Inango HPNA equipment under iMMS.

Other license file types are usually custom-prepared and may be for demo systems or for creating provider-specific restricted networks. Please consult your Inango sales representative for further details.

License files may be field-updated (though by nature this is usually a rare operation). Please refer to the following section on Software Upgrade for a description of how licenses may be upgraded.

3.6 Software Upgrade

iMMS is fully field-upgradeable. The software upgrade operation is controlled by the "copy" function, whose parameters are kept in the copyTable. The upgrade process

preserves managed data that was prepared in under the current (or previous) software image.

Note: The iMMS system will automatically reboot after a software upgrade.

Depending on the hardware platform on which iMMS is running, a copy of the previous software image is usually kept alongside the current active image. If the upgrade process fails unexpectedly (e.g. from a power outage) then the previous (and still intact) image will be used.

There is also a copyHistoryTable, where information from previous copy operations may be found.

The copy commands are a generic copy operation that may be used to upgrade iMMS software, upgrade HPNA device firmware, download new license files and import/export of configuration data. This section relates to their use for upgrading software only.

In order to perform a software (or iMMS license) upgrade operation via the web interface (on the System Info → Copy page), press the radio button on the "Add new entry" row and then set the following parameters in the copyTable:

Operation ID	A unique integer value that identifies this copy operation in the copyTable and copyHistoryTable (web and SNMP interfaces only: for web, it is automatically generated)
Source Protocol	Protocol (ftp, tftp,) to be used to download the new image
Source Host Name	Name or IP address of the remote station from which the new image is to be downloaded
Source File Name	Name of file on remote station containing the new image
Destination File Type	Either "image" for software upgrade or "license" for upgrading the iMMS license
Status	"noop": operation ready to start but not yet started; "applying": downloading or verification in process. You may still abort the operation. "pending": new image is being burned to flash. DO NOT ABORT THE OPERATION.
Abort	Set to "true" to abort the current upgrade operation (must be "false" to start the upgrade)

Once the parameters have been set, press the "Apply" button to perform the copy operation.

Information kept in the copyHistoryTable is as follows:

Operation ID	Identifies the operation, as defined in the copyTable
Time	Time and date on which upgrade completed
Status	“success” or “failure”
Description	IMMS version number (if upgrade was successful) Reason for upgrade failure if the operation failed

Notes that are specific to the front-end you are using with iMMS:

– in the HHTP interface, the copyTable is found under the SystemInfo tab

– when using SNMP, open a row new with status “CreateAndWait” in the copyTable (OID: 1.3.6.1.4.1.40354.1.1.1.1.1.1) . Set the rowStatus to “Active” once the parameters have been set.

– In the CLI, the copy command is located at the “Console” level. It's syntax is as follows:

`tftp://<ipaddr>/<filename> image` (to upgrade the software)

`tftp://<ipaddr>/<filename> license` (to upgrade iMMS license)

3.6.1 Automatic code upgrades

Version 10.8 includes an automatic upgrade feature for both software (iMMS) and the HPNA firmware for both master and endpoint units.

In order to upgrade the software and/or firmware of the network, the iMMS must be first configured to check for upgrades periodically at an upgrade server . The upgrade server is a location at which information about software upgrades can be found by iMMS (the code files themselves may be elsewhere).

Once there is new code for upgrade, iMMS will download it as appropriate. If endpoint firmware is upgraded, then it will be applied to all endpoints attached to the iMMS system. If an endpoint was detached at the time that the auto-upgrade is run and at a later point in time goes on-line, the endpoint will be upgraded automatically.

The details of the auto-upgrade system are as follows:

– The auto-upgrade server as specified in the iMMS configuration for auto-upgrades) is just the location of a computer with an HTML file of the format below. It is read by the iMMS (via either HTTP or TFTP) in order to locate specific code files to upgrade.

Image Type	Type of image: SW (iMMS software), firmware for masters or firmware for endpoint units
File name	Name of file containing image (e.g. new-fw.bin)
Link to file	URL of file (e.g. http://10.0.0.1/new-fm.bin)
Image signature	MD5 signature of image
Version number	Version of new code
Info signature	Reserved for future use
Comments	General comments

– iMMS periodically reads the above file and checks if new code is available
NOTE: Code must be used in the format as supplied by Inango.

3.7 Managing Users

Users of the iMMS system may be defined and assigned different levels of authorization.

There are three levels of authorization for iMMS users:

Admin	Full read/write access to all tables
Config	Read/write access to “config” tables only. “admin” tables are not accessible
Review	Read-only access to “config” tables only

3.7.1 Create a New User

User names must begin with a letter and then may contain letters, digits, underscores or hyphens. User names are case-sensitive.

Users are added in the userManagementTable.

The CLI command for adding a new user is:

```
hpna/mngm/user-manag> config <user> password <passwd> group <accessPerms>
        where <access> can be: 1 (review), 2 (config), 3 (admin)
```

Delete the user by typing:

```
hpna/mngm/user-manag> config no <user>
```

Modify the user by typing:

```
hpna/mngm/user-manag> config <user> password <new-passwd> group <new-accessPerms>
```

3.8 Network Interfaces

Network interfaces in iMMS are either a physical interface (referred to as a “nic”) or a VLAN (“vlan”). Each interface is defined in the interfaceTable.

Also related to interfaces is the ipAddressTable which is used to define addresses, netmasks, etc for each system interface. Address types may be “dhcp” or “static”. If DHCP is specified for any row in this table, you can not set the IP address, netmask or default gateway address.

Factory defaults are 192.168.1.2 for first interface (and 192.168.2.2 for second and so on if they exist). Default netmask is 255.255.255.0. Default gateway is the same as the default IP address for this interface.

If you change an address type to “dhcp”, iMMS will load new IP address, netmask and default gateway values from a DHCP Server. If no DHCP Server is found on the network, it will revert to the previous values. Changes are recorded in syslog (if activated).

3.9 Authentication

IMMS supports authentication via 802.1x.

(a) If 802.1x authentication is enabled for the master unit, it first needs to be authenticated within the network via 802.1x. The master (running iMMS) sends 802.1x authentication requests (as an 802.1x Supplicant) and is duly authenticated for the network. Only after it has been authenticated will the master be able to connect to the network.

(b) CPE equipment in a user's apartment and that is connected to an HPNA endpoint can be authenticated with a central Radius Server via the 802.1x protocol. In this case, the master unit (running iMMS) acts as an 802.1x Authenticator (it translates the incoming 802.1x authentication requests in to Radius requests and sends to the Radius server for handling. In this process, the user CPE equipment will be authenticated.

Note that when enabling 802.1x in iMMS, there are two distinct flags that must be set – one for Supplicant and one for Authenticator mode.

3.10 Useful Functions

After basic configuration of the iMMS station itself, some of the functions that may be of interest are summarized below (full details are documented in later sections of this manual).

3.10.1 Topology

The Topology section (in the web interface, this is under the “System Info” tab) shows all equipment seen by the iMMS (the data here is read-only). There are three layers here: HPNA Masters, HPNA Endpoint and User CPE Equipment.

If iMMS is running on an HPNA Master (a “managed master”) then the masters topology is the unit on which iMMS is running; otherwise (i.e. iMMS is running on a computer), all master units on the network are displayed. The masters in the topology are designated by the MAC address on their HPNA interface.

The Endpoints topology is the list of endpoints. They are also identified by their HPNA MAC address and the master to which they are attached is identified by it's HPNA MAC.

The third part of the topology screen is the Customer Topology. These are the user devices connected to an endpoint in the user's apartment. If you have enabled DHCP Relay, then the allocated IP address of each device is displayed. In any event, the MAC address of the user device (on its Ethernet interface that is connected to the HPNA Endpoint) is displayed, along with its status and the HPNA MAC of the endpoint unit.

3.10.2 Testing Line Quality (Diagnostics)

A common request is to check line quality of the coax for HPNA operations. iMMS supports two types of line diagnostics tests: full and light-weight diagnostics. Both tests measure signal-noise ratio (SNR) and up/down-stream data and PHY rates. Full diagnostics tests also measure Packet-Error Rate (PER).

Full diagnostics are from a master unit (from the “Masters” tab). Line quality is then measured for all endpoints that are connected to this master over the coaxial network. The results of the full diagnostics tests may be read at the endpoint's tables (in the “Endpoints” tab) – but note that these tables are read-only (i.e. the tests are run from the master only).

Light-weight diagnostics tests are run at the endpoint-level for any individual endpoint. Measured results are the same with the exception of PER which is not measured in the light-weight tests.

Full diagnostics tests are time-consuming and while running will prevent any user data from being passed over the coaxial network. Expect the tests to take approximately 30 seconds plus 10 seconds for each endpoint connected to the master until they complete.

3.10.3 Cross-Talk

Cross-talk occurs when a master “hears” other masters on the coaxial interface. This is detected by the cross-talk test and usually indicates a cabling problem in the network.

3.10.4 Noise Test

This run the noise test as supported by the HPNA devices. For full meaning of these results, please consult the HPNA documentation.

3.10.5 Configuring Rate Limitation

An important part of configuring your HPNA network is traffic management. IMMS allows you to limit up/down-stream traffic to/from each endpoint. This can be used to share the bandwidth in the desired way among all endpoints on the HPNA network. If you are using the web interface, the rate limitation screens are located under the “Endpoints” tab.

3.10.6 Multi-Cast Operations

In order to configure multi-cast operations, you will usually need to enable IGMP Snooping on the master unit (see the “Multicast Master Table” under “Multicast info” in the “Masters” tab).

Note: Make sure you set the timeout value correctly for your network: if it is not, IGMP will most likely not work properly.

You can then define multi-cast parameters for each endpoint (under the “Endpoint” tab).

3.10.7 Link Aggregation

By enabling link aggregation (LAG) on an HPNA Master, you effectively combine the two 100Mbps Fast-Ethernet ports in to a single logical interface of 200Mbps (you also need to enable LAG on the switch to which the Master is connected).

LAG is enabled for each Master individually (see the Master Table) and requires no specific configuration information.

There are a few small restrictions to LAG in iMMS:

- LACP is not supported (LAG is statically configured)

- upstream traffic (from the Master to the network) is always transmitted over one interface
- If the upstream interface is disconnected, the Master will switch to the second interface within 3 seconds

3.10.8 DHCP Relay

DHCP Relay allows the iMMS to intercept DHCP traffic and track IP addresses allocated by the DHCP Servers to end-user equipment. It must be enabled globally (at the iMMS level) and then activated on each HPNA Master for which the functionality is required.

In order to configure DHCP Relay, we need to define the actual DHCP Servers that will be used and the DHCP Relay options for each HPNA Master using it. The relevant options are the Remote-ID and Circuit-ID fields in the DHCP Relay packets sent to the server. These are defined in the per-master information table in the iMMS tab.

Note: User-defined values (for Remote-ID and Circuit-ID) are not currently supported. Supported values for each field are Endpoint HPNA MAC and Master HPNA MAC.

3.10.9 Filters

Filters may be used with either HTTP or CLI interfaces to only display information matching supplied criteria.

The syntax for defining filters is as follows:

```
show filter "KEYWORD1 EXPRESSION1 KEYWORD2 EXPRESSION2 …"
```

where

KEYWORD specifies the parameter for filtering.

Supported keywords are:
interface_id, master_mac, master_status, master_alias, master_swver, master_auth, master_lag, master_igmp, master_dhcp, ep_mac, ep_status, ep_alias, ep_swver, ep_auth, ep_igmp

EXPRESSION = [RELATION] VALUE
Valid relations are <, >, =, !. Empty relation is regarded as equality (=)
Value can contain wildcards (*) and (?).
Star (*) is equal to any string including empty string ("").
Question mark equals exactly one arbitrary character.

EXAMPLES:

1. Show all online EPs connected to master specified by MAC

```
hpna ep info show filter "master_mac 00:11:22:33:44:55 ep_status 1"
```

2. Show all masters connected to iMMS with specified interface

```
hpna master info show filter "interface_id = 1"
```

3. Show all masters having EP with specified MAC prefix

```
hpna master info show filter "ep_mac 00:11:22:*"
```

4. Managed Data

iMMS divides managed data in to four main groups, each of which is sub-divided in to several smaller groups of information. In this section, we detail the managed data elements in our system.

1. System Group (1.3.6.1.4.1.40354.1.1.1.1.1)

   The System Group maintains system-wide information. Within this group, we can activate functions such as discovering the topology of the entire HPNA network, backing up system configuration, software upgrades and resetting system values to factory defaults.

   Sub-groups in the System Group are defined a follows:

   - copyTable (1.3.6.1.4.1.40354.1.1.1.1.1.1)

   Each row in the copyTable defines a new “copy” operation that is to be activated. When the copy operation finishes, the row is automatically deleted from the table.

   The copy operation is a high-level copy that transfers data from any endpoint, master or local station via a range of “copy protocols”: , tftp or copy from memory. The data copied may be either a unit's configuration, firmware (HPNA software), image (application code), boot code or the iMMS license.

   Alongside the copyTable is the updateHistoryTable, in which we store results of previous copy operations.

   - genericParamsGroup (1.3.6.1.4.1.40354.1.1.1.1.1.2)

   This group is used to perform global operations such as rebooting the iMMS station, restoring HPNA parameters to system defaults or restoring all values to system defaults.

   - topologyGroup (1.3.6.1.4.1.40354.1.1.1.1.1.3)

   The topology group stores the topology for the entire HPNA network. There are three tables in this group. The first is a table of master units that the management system has discovered over all NIC interfaces and VLANs. Note that any individual master may be visible over several NIC/VLANs – each will be listed separately in this table.

   The second table in the topology group is the list of endpoints connected to each master.

   The third table in this group is called the “customer” topology: MAC addresses of devices (computers, etc) in the user's home LAN that are connected to his HPNA endpoint. If DHCP Relay is enabled (see “DHCP Relay” under the “iMMS Info” tab), then the allocated IP addresses of these user devices is also displayed.

2. Management Group (1.3.6.1.4.1.40354.1.1.1.1.2)

The Management Group information relates to the iMMS station itself (in the web interface, this is under the “iMMS Info” tab). Note that the management system may run on either a PC/Linux platform connected to the masters (via Ethernet) or on one of the master units itself (provided it is a managed master). Parameters in the Management Group are used to configure the management software in both cases.

Note: Users of previous versions will note that the information in this tab has been considerably extended.

Information in the iMMS Info tab is arranged in the following sections:

- interfaceTable (1.3.6.1.4.1.40354.1.1.1.1.2.1)

This defines the network interfaces on the management system. An interface is either a VLAN or a NIC card: via this table, you may assign a name to the interface and change its status. Use *adminStatus* to set the interface to either “up” or “down” (the current status of the interface is always reflected in *operationStatus*).

- ipAddressTable (1.3.6.1.4.1.40354.1.1.1.1.2.2)

IP-related information for the management system (DHCP or static IP address, etc). The management system currently supports IPv4 only.

- iMMSVlanGroup (1.3.6.1.4.1.40354.1.1.1.1.2.3)

This table defines VLAN's over which the iMMS is accessed . A VLAN Table is used to define VLAN values and aliases, the VLAN Port Table defines which VLAN tags are used on which ports.

- userManagmentTable (1.3.6.1.4.1.40354.1.1.1.1.2.4)

This table is used to define users that may access the iMMS system. Each user is given a “permission” (review, config or admin) that defines his rights with respect to various parts of the system (admin has full write permission for all fields).

- iMMSDiagnosticGroup (1.3.6.1.4.1.40354.1.1.1.1.2.5)

iMMS diagnostics are network-level tests that may be used in test for connectivity, basic response times, etc. Tests supported are ping, traceroute and ARP.

- iMMSSettings (1.3.6.1.4.1.40354.1.1.1.1.2.6)

The iMMS Settings define lower-level configuration parameters that are specific to the front-end you are using to access iMMS (WEB, SNMP or CLI).

For web users, this parameter may be used to display tables either horizontally (as they are in previous versions) or vertically. This parameter applies to all tables in the system.

- iMMSHostname (1.3.6.1.4.1.40354.1.1.1.1.2.9)

Hostname of the iMMS station

- timeGroup (1.3.6.1.4.1.40354.1.1.1.1.2.10)

This is used to configure the time settings for iMMS: time-zone, format for time and date display and either enabling NTP client or defining local time. If NTP is defined,

the time-zone (and not the time) may changed; if NTP is not being used, then the time (and not the time zone) may be changed.

If you enable the NTP Client, use the NTP Servers table to define the NTP Server(s).

- iMMSLicenseInfo (1.3.6.1.4.1.40354.1.1.1.1.2.11)

Use of iMMS is governed by the license file with which it is supplied: this is used (among other things) to allow service providers to protect their equipment from use in other networks. Questions on your licensing rights should be directed to your Inango sales representative.

Important note: You cannot change the licensing information. Any attempt to do so may render the unit useless.

- iMMSSWRelease (1.3.6.1.4.1.40354.1.1.1.1.2.12)

The version of the iMMS software.

- primaryDNSServer (1.3.6.1.4.1.40354.1.1.1.1.2.14)

The primary DNS Server for the iMMS station

- secondaryDNSServer (1.3.6.1.4.1.40354.1.1.1.1.2.15)

The secondary DNS Server for the iMMS station

- defaultGateway (1.3.6.1.4.1.40354.1.1.1.1.2.16)

The Default Gateway for the iMMS station

- iMMSDhcpRelayGroup (1.3.6.1.4.1.40354.1.1.1.1.2.17)

DHCP Relay is a new feature added in release 10.5. It allows network administrators to track IP addresses allocated to end users on the network (it is a functional super-set of DHCP Option 82).

When enabling DHCP Relay, iMMS will intercept all DHCP traffic from end users, record information as required and then send the DHP traffic to pre-defined DHCP Servers (via the DHCP Relay protocol).

- iMMSHWInfo(1.3.6.1.4.1.40354.1.1.1.1.2.18)

The hardware information related to the station on which iMMS is running.

- iMMSRoutingGroup (1.3.6.1.4.1.40354.1.1.1.1.2.19)

Routing information is the routing table (next hops, etc) and ARP table for each interface on the iMMS' station.

3. Masters Group (1.3.6.1.4.1.40354.1.1.1.1.3)

   The Masters Group manages the MDU Masters in the network. There are three sub-divisions of data in this group:

   - masterInfoTable (1.3.6.1.4.1.40354.1.1.1.1.3.1)

Information specific to each individual master: MAC address (on HPNA interface), model number, vendor-specific information and so on.

Two fields in this table are not self-explanatory: “Authorized” indicates whether this equipment is authorized Inango equipment and “LAG” is used to enable/disable LAG (Link Aggregation) functionality on this master.

- masterStatisticGroup (1.3.6.1.4.1.40354.1.1.1.1.3.3)

This is a large group of statistics gathered for each MDU master. Statistics are recorded (i) count over the past 1 hour, (ii) count over the past 12 hours, (iii) count over the past 24 hours, (iv) last read statistic and (v) aggregated since initial installation of the unit (survives resets and power-downs).

Statistics are gathered over each interface (HPNA and each Ethernet port) separately.

- masterMacMcastGroup (1.3.6.1.4.1.40354.1.1.1.1.3.4)

Multi-cast handling. For the Master units, there are only two fields to set for multi-cast and IGMP handling: an enable field for IGMP snooping and the IGMP snooping timeout.

Note: The IGMP timeout value must be configured to match the same values in other parts of the network in order for IGMP to function properly. CHECK THE IGMP SETTINGS ON YOUR SWITCH.

4. Endpoints Group (1.3.6.1.4.1.40354.1.1.1.1.4)

   There are six sub-divisions of data (tables and smaller groups) in the Endpoints Group.

   - epInfoTable (1.3.6.1.4.1.40354.1.1.1.1.4.1)

   Information specific to each individual endpoint: MAC address (on HPNA interface), model number, vendor-specific information and so on. The “Authorized” field indicates whether this unit is an authorized Inango-supplied unit or not.

   - epDiagnosticGroup (1.3.6.1.4.1.40354.1.1.1.1.4.2)

   Run and read results of diagnostics tests using tables in the epDiagnosticGroup. There are two generic types of diagnostics tests: full tests and lightweight tests. Full tests calculate line parameters Packet-Error Rate (PER), Signal-Noise Ration (SNR), data rates, power and PHY rate. All of these are calculated separately for upstream and downstream directions. ***Full diagnostic tests are traffic-affecting***.

   Lightweight diagnostics calculate the same values (upstream and downstream) except the PER calculations. Light diagnostics tests are ***NOT*** traffic affecting.

   - epStatisticGroup (1.3.6.1.4.1.40354.1.1.1.1.4.3)

   This is a large group of statistics gathered for each endpoint. Statistics are recorded (i) count over the past 1 hour, (ii) count over the past 12 hours, (iii) count over the past 24 hours, (iv) last read statistic and (v) aggregated since initial installation of the unit (survives resets and power-downs).

   Statistics are gathered over each interface (HPNA and each Ethernet port) separately.

- epMcastGroup (1.3.6.1.4.1.40354.1.1.1.1.4.4)

Multi-cast handling. For the Endpoint units, there are only two fields to set for multi-cast and IGMP handling: an enable field for IGMP snooping and the IGMP snooping timeout.

- epRateLimitationGroup (1.3.6.1.4.1.40354.1.1.1.1.4.5)

Upstream and downstream rate limitation parameters (entered in Kpbs).

Limits are provided for: upstream Unicast traffic (epHPNAUCUpstreamLimit), upstream broadcast traffic (epHPNABCUpstreamLimit), upstream multicast traffic (epHPNAMCUpstreamLimit) and general downstream traffic.

In the upstream direction, the limitations may be used to restrict the amount of each type of traffic this unit transmits on the HPNA coaxial network. Downstream limitations define the rate at which Ethernet packet are transmitted on the local (in-home) network.

- epMacGroup (1.3.6.1.4.1.40354.1.1.1.1.4.7)

(This is the “Network” link in the web interface).

This defines options related to VLANs on this endpoint. Note that VLANs are defined via two sub-tables in this group: they may be defined (and tagged, untagged) at the HPNA or Ethernet interfaces on the endpoint. Separate tables are used in each case.

When defining VLAN handling for the HPNA (RF) interface, field that must be set are an enable field, a VLAN-id (tag) and a boolean value that defines whether other VLANs are pass through unchanged.

VLAN's on the endpoints' Ethernet interfaces are defined by the VLAN-id (tag) and boolean values for enabling/disabling port tagging and whether to remove VLAN tags or not.

5. Command-Line Interfaces

All managed data is also accessible via a Command-Line Interface (CLI). The commands are organized in a tree-like structure that closely follows the tables outlined in the previous sections. They use a Cisco-like interface (implementation is based on the “clish” package).

The command “directories” in the tree are as follows:

`copy`				
`ping`				
`traceroute`				
`hpna`				
	`system`			System-level operations
		`topology`		Topology-related commands
			`if-master`	Interfaces on masters
			`master-ep`	Endpoints connected to masters
			`customer-ep`	Customer equipment connected to endpoints
		`Update-history`		History of copy commands and results
	`mngm`			iMMS configuration and operations
		`interface`		Interfaces on mng system (NIC or VLAN)
		`ip`		IP configuration data (address, gateway, netmask, etc)
		`vlan`		iMMS VLAN's
			`vlan-if`	Defined VLAN's
			`vlan-port`	Per-port options for VLAN's

		user-manag		User names and passwords
		diag		IMMS-level diagnostics
		iMMS		Per-front-end commands
			snmp	Global SNMP configuration
			log	Global log configuration
		time	ntp-server	Time (NTP) options
		dhcp-relay		DHCP Relay configuration
			server-info	DHCP Server configuration
			master	Per-Master unit config for DHCP Relay
		route		Routing commands
			rt	Routing table
			arp	ARP table
	master			MDU/Master units
		info		Per-master information
		statistics		Statistics (per port and per-master)
		multicast		Multi-cast options
	ep			MDU Endpoint information
		info		Per-endpoint information
		diagnostic		Diagnostics tests
		statistics		Statistics (per-port and per-endpoint)
		multicast		Multi-cast options
		rate-limit		Rate Limitation
		mac		MAC (layer 2) configuration:

				MAC address, VLAN options, etc
			hpna	On HPNA interface
			net	On Ethernet interfaces

5.1 Deleting entries in config tables

All of the “config” commands are used to configure tables. This means either adding a new row in a table, editing an existing row or deleting a row. In order to delete a row from a table, the reserved word “no” can be used as follows:

The following command is used to config a row in the iMMS interface table:

```
hpna mngm interface config <interface> {parameters...}
```

To delete a row from the interface table, type the following:

```
hpna mngm interface config no <interface>
```

The “no” operator may be used in any of the “config table” commands:

5.2 Command list

Commands available at the CLI are as shown below. Those commands noted with (*) are documented in greater detail in the later chapter of this document.

```
copy <source-url> <dest-url> (*)

ping <destination> [size packet_size] [count packet_count] [timeout
time_out] (*)

traceroute <destination> [ttl max-ttl] [count packet_count] [timeout
time_out] [tos tos] [packet_size] (*)

hpna system topology if-master show

hpna system topology master-ep show

hpna system topology customer-ep show

hpna system topology customer-ep show

hpna system update-history show

hpna system update-history show

hpna mngm interface show

hpna mngm ip show
```

```
hpna mngm vlan vlan-if show
hpna mngm vlan vlan-port show
hpna mngm user-manag show
hpna mngm imms snmp trap-srv show
hpna mngm imms log local show
hpna mngm imms log local show
hpna mngm imms log syslog-server show
hpna mngm time ntp-server show
hpna mngm dhcp-relay server-info show
hpna mngm dhcp-relay master show
hpna mngm route rt show
hpna mngm route arp show
hpna mngm auth supplicant show
hpna master info show
hpna master diag rf show
hpna master diag xtalk show
hpna master diag noise-test show
hpna master statistics rf show
hpna master statistics dev-info show
hpna master statistics dev-info show
hpna master multicast master-mcast show
hpna ep info show
hpna ep diagnostic hpna show
hpna ep diagnostic hpna-lite show
hpna ep diagnostic noise-test show
hpna ep statistics rf show
hpna ep multicast ep-mcast show
hpna ep rate-limit hpna show
hpna ep mac hpna show
hpna ep mac net show
hpna mngm interface config no <interface>
```

```
hpna mngm interface config <interface> [type <value> ] [alias <value> ]
[name <value> ] [operation <value> ] [management <value> ] [hpna <value>
] [mac <value> ] (*)

hpna mngm ip config no <interface> (*)

hpna mngm ip config <interface> [address <value> ] [mask <value> ] [prio
<value> ] [type <value> ] (*)

hpna mngm vlan vlan-if config no <vid>

hpna mngm vlan vlan-if config <vid> [alias <value> ]

hpna mngm vlan vlan-port config no <vlan-tag> <port-ifindex>

hpna mngm vlan vlan-port config <vlan-tag> <port-ifindex> [enable-tag
<value> ] (*)

hpna mngm user-manag config no <user>

hpna mngm user-manag config <user> [password <value> ] [group <value> ]

hpna mngm imms snmp trap-srv config no <id>

hpna mngm imms snmp trap-srv config <id> [host <value> ] [prio <value> ]

hpna mngm imms log syslog-server config no <id>

hpna mngm imms log syslog-server config <id> [ip-addr <value> ] [enable
<value> ]

hpna mngm time ntp-server config no <id>

hpna mngm time ntp-server config <id> [ip <value> ] [prio <value> ]

hpna mngm dhcp-relay server-info config no <id>

hpna mngm dhcp-relay server-info config <id> [address <value> ] [prio
<value> ]

hpna mngm dhcp-relay master config no <master-mac>

hpna mngm dhcp-relay master config <master-mac> [enable-relay <value> ]
[remote-type <value> ] [remote-string <value> ] [circuit-type <value> ]
[circuit-string <value> ] (*)

hpna mngm route rt config no <entry-id>

hpna mngm route rt config <entry-id> [dest <value> ] [mask <value> ]
[next-hop <value> ] [if <value> ] [metric <value> ] (*)

hpna mngm route arp config no <entry-id>

hpna mngm route arp config <entry-id> [dest <value> ] [mac <value> ] [if
<value> ] (*)

hpna master info config no <master-mac>
```

```
hpna master info config <master-mac> [reboot <value> ] [reset <value> ]
[reset-subsystem <value> ] [lag <value> ] (*)

hpna master diag rf config no <master-mac>

hpna master diag rf config <master-mac> [start-test <value> ] [number-
packets <value> ] [reset <value> ] [reset-ligth <value> ]

hpna master diag xtalk config no <master-mac>

hpna master diag xtalk config <master-mac> [start-test <value> ] [clear
<value> ]

hpna master diag noise-test config no <master-mac>

hpna master diag noise-test config <master-mac> [start-test <value> ]
[reset-res <value> ] [iterations <value> ]

hpna master statistics rf config no <master-mac>

hpna master statistics rf config <master-mac> [clear <value> ]

hpna master multicast master-mcast config no <master-mac>

hpna master multicast master-mcast config <master-mac> [igmp <value> ]
[igmp-timeout <value> ] (*)

hpna ep info config no <ep-mac>

hpna ep info config <ep-mac> [reboot <value> ] [alias-name <value> ]
[reset <value> ] [enable-traffic <value> ] [host-number <value> ] (*)

hpna ep diagnostic hpna-lite config no <ep-mac>

hpna ep diagnostic hpna-lite config <ep-mac> [start-test <value> ]
[number-packets <value> ]

hpna ep diagnostic noise-test config no <ep-mac>

hpna ep diagnostic noise-test config <ep-mac> [start-test <value> ]
[reset-res <value> ] [iterations <value> ]

hpna ep statistics rf config no <ep-mac>

hpna ep statistics rf config <ep-mac> [clear <value> ]

hpna ep multicast ep-mcast config no <ep-mac>

hpna ep multicast ep-mcast config <ep-mac> [igmp <value> ] [igmp-timeout
<value> ]

hpna ep rate-limit hpna config no <ep-mac>

hpna ep rate-limit hpna config <ep-mac> [unicast-upstream <value> ]
[broadcast-upstream <value> ] [multicast-upstream <value> ] [downstream
<value> ] [downstream-peak <value> ]

hpna ep mac hpna config no <ep-mac>
```

```
hpna ep mac hpna config <ep-mac> [vlan <value> ] [tag <value> ] [other-
vlan-drop <value> ]
hpna ep mac net config no <ep-mac> <port>
hpna ep mac net config <ep-mac> <port> [vlan <value> ] [port-taged
<value> ] [tag-remove <value> ]
hpna system param show
hpna system param auto-up show
hpna mngm imms snmp show
hpna mngm imms web show
hpna mngm imms log show
hpna mngm show
hpna mngm time show
hpna mngm dhcp-relay show
hpna mngm auth authenticator show
hpna mngm auth show
hpna master statistics show
hpna ep diagnostic show
hpna ep statistics show
hpna system param config reset <value>
hpna system param config reboot <value>
hpna system param config hpna-reset <value>
hpna system param config safe-reset <value>
hpna system param config auto-upgrade <value>
hpna system param auto-up config time-interval <value>
hpna system param auto-up config transaction-interval <value>
hpna system param auto-up config web-page <value>
hpna system param auto-up config enable <value>
hpna mngm imms snmp config public-com <value>
hpna mngm imms snmp config private-com <value>
hpna mngm imms snmp config sys-location <value>
hpna mngm imms snmp config sys-contact <value>
hpna mngm imms web config line-per-table <value>
hpna mngm imms web config info-direction <value>
```

```
hpna mngm imms log config logfile-max-level <value>
hpna mngm imms log config logfile-max-msg <value>
hpna mngm imms log config logfile-en <value>
hpna mngm imms log config logfile-clean <value>
hpna mngm imms log config syslog-max-level <value>
hpna mngm config hostname <value>
hpna mngm config license <value>
hpna mngm config release <value>
hpna mngm config primary-dns <value>
hpna mngm config secondary-dns <value>
hpna mngm config default-gw <value>
hpna mngm config hw-info <value>
hpna mngm time config zone <value>
hpna mngm time config format <value>
hpna mngm time config local <value>
hpna mngm time config ntp-en <value>
hpna mngm dhcp-relay config enable-relay <value>
hpna mngm auth authenticator config eap-reauth-period <value>
hpna mngm auth authenticator config eapol-version <value>
hpna mngm auth authenticator config ip-addr <value>
hpna mngm auth authenticator config nas-identifier <value>
hpna mngm auth authenticator config prim-auth-server-addr <value>
hpna mngm auth authenticator config prim-auth-server-port <value>
hpna mngm auth authenticator config prim-auth-server-shared-secret
<value>
hpna mngm auth authenticator config sec-auth-server-addr <value>
hpna mngm auth authenticator config sec-auth-server-port <value>
hpna mngm auth authenticator config sec-auth-server-shared-secret
<value>
hpna mngm auth authenticator config acct-server-addr <value>
hpna mngm auth authenticator config acct-server-port <value>
hpna mngm auth authenticator config acct-server-shared-secret <value>
hpna mngm auth config enable-auth <value>
```

```
hpna mngm auth config enable-suppl <value>
hpna master statistics config reset-stats <value>
hpna ep diagnostic config reset-diag <value>
hpna ep diagnostic config reset-light-diag <value>
hpna ep diagnostic config reset-noise <value>
hpna ep statistics config reset-stats <value>
```

6. HTTP Interfaces

This section describes the web screens that can be used to operate iMMS.

6.1 Overview

The web interface is divided into several functional zones that combine operations and/or information of a related nature. You can navigate between these areas via the tabs at the top of the screen. These are explained in further detail in this chapter.

After navigating to a particular zone, parameters of relevance are shown in the center of the page. The specifics will vary from one screen to another, but may include such as hostname, time, etc.

On the left-hand side of the management panel there is a list of tables available for the reading and modification in the selected zone. Click on one of them to view an appropriate table, which will then be displayed at the right of the list.

In order to modify information in a table, the user must first choose select a row by clicking one of the radiobuttons in the left-most column of the table. Some tables allow to you to create new rows, in such cases the last row is named "*Add new row*" and you can select it to add a new row to that table.

If table supports row deletion, there is a "*Delete*" button, which deletes the row from the database. This also activates an appropriate deconfiguration process in the system (i.e. Remove data base entries and performs systems actions resulting in the removal of the row).

To edit an existing row, select it and its values are then displayed at the bottom of the table. You can modify these values (those that are not read-only) and once ready, press the "*Apply*" button in order to apply your changes.

If you wish to quit the session for any reason (for example, in order to downgrade/upgrade permissions by changing the user name), press the "Logout: user" link at the right of the main tab panel.

6.2 Filters

Many of the web pages in iMMS display tables of information. In most cases, it is possible to define “filters” that can be used to restrict the size of the displayed table to only those parts that are of relevance.

Where filters are used, there is a button on the screen called “Show Filters” (or Hide Filters) that, when pressed, opens a table of filter parameters. These parameters include logical expressions, wild-cards characters, etc – only table rows matching these filters will be displayed.

For a more complete description of the logical expressions, reserved words and wild-cards, please refer to the description of filters in Chapter 3 above.

6.3 Login

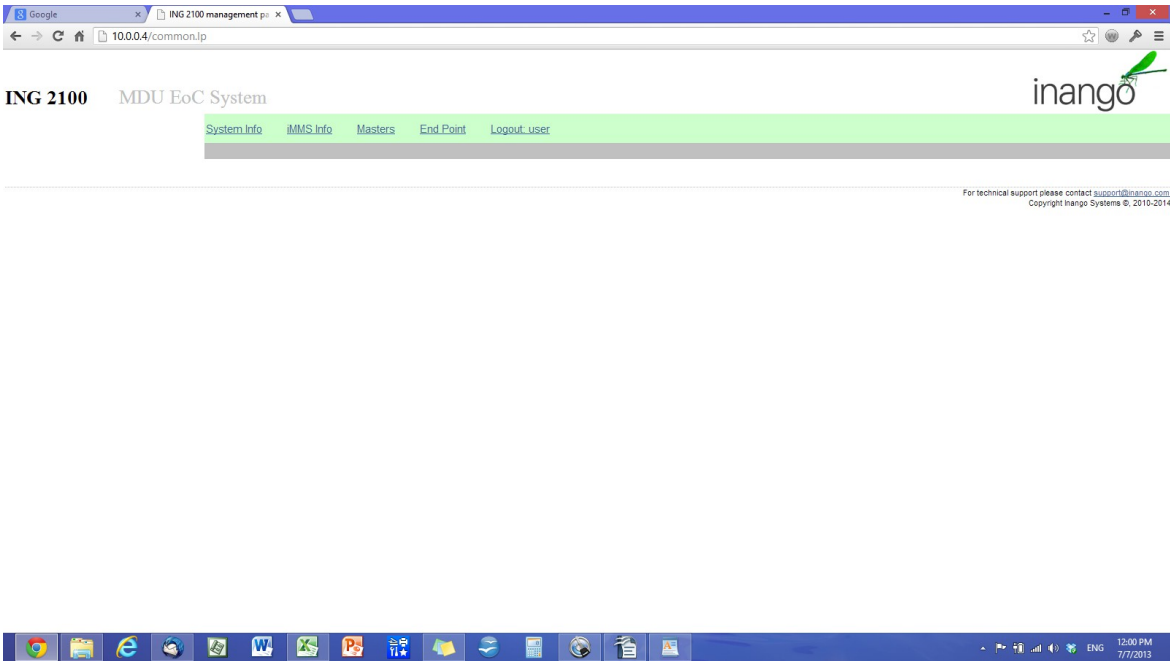
Enter user name and password to enter the iMMS system. Each user belongs to a *review*, *config* or *admin* group. Members of a *review* group may read the system's data base, members of the *config* group can edit managed data settings and the *admin* group is used to perform all operations (including user management – see the CPU Info => User Management Table).

Note: iMMS is shipped with a default admin user “user”, password “changeme”.

Note that if the system is inactive for a period of time, you will be automatically logged out. This is a security feature and does not affect any prior operations you have done with iMMS.

6.4 Opening Screen

The opening screen provides the initial navigation choices: System Info (update software, display system topology, etc), iMMS Info (setting related settings), Masters, End Point and Logout.



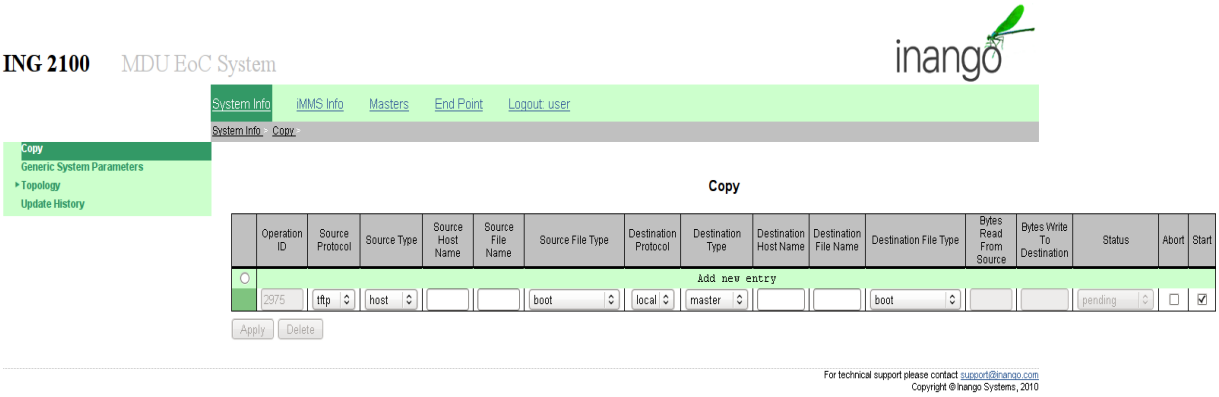
6.5 System Info Tab

This tab contains tables that provide information and functions relating to the network system administration functions necessary to manage the HPNA network. These include functions for updating software and data bases, viewing the entire HPNA network topology, general resets and “return to factory defaults”.

6.5.1 Copy

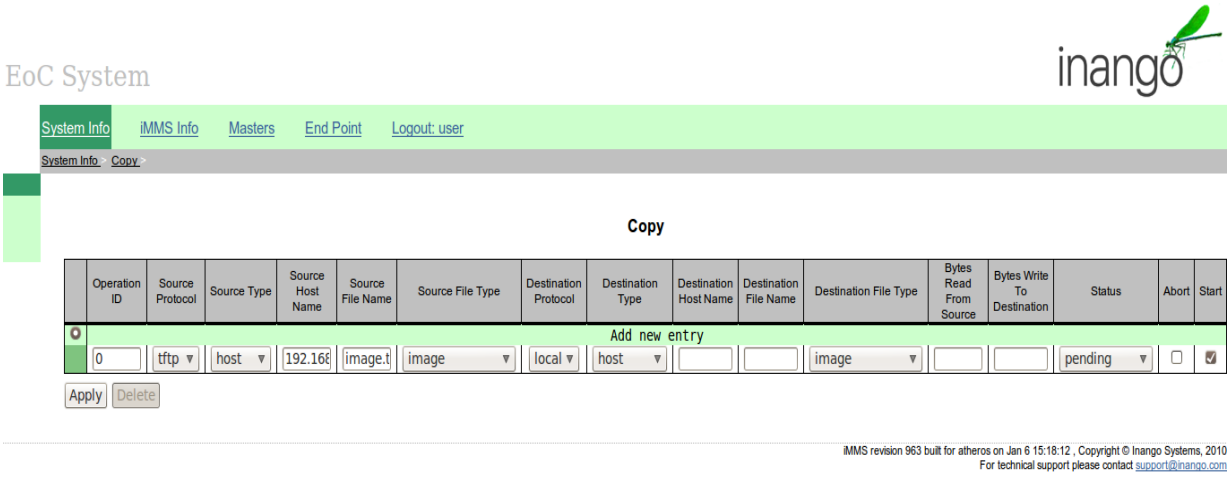
The “Copy” screen is used to update (download) iMMS software, HPNA units' firmware, configuration data base and licensing information files. Additionally, it can be used to upload the data base files. The source or destination of the copy operation may be any computer, the iMMS station, or any HPNA Master or HPNA Endpoint unit on the network.

If you click on the “copy” bar under “System Info”, a screen of options for the copy command will appear.

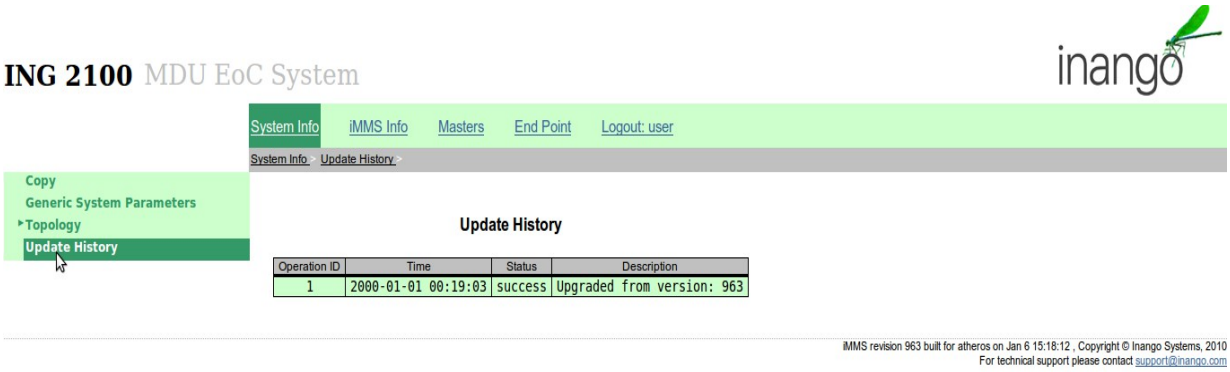


The various options for the “copy” command are described in fuller detail in the chapter below (see page 73). Briefly, though, we can copy files of various types: “image” is the iMMS software image, “firmware” is HPNA firmware running on either a master or endpoint unit, “license” is the iMMS license file, “runningconfig” is the current configuration data base and so on. We need to define a source and destinations for the copy operation (a computer on the network, the iMMS station itself, an HPNA master or endpoint unit) and a protocol (TFTP).

As an example, if we wish to upgrade the iMMS software image from an image file (supplied by Inango) named image.t that is located on a computer with IP address 192.168.100.100 then the copy operation parameters should be filled out something like below:

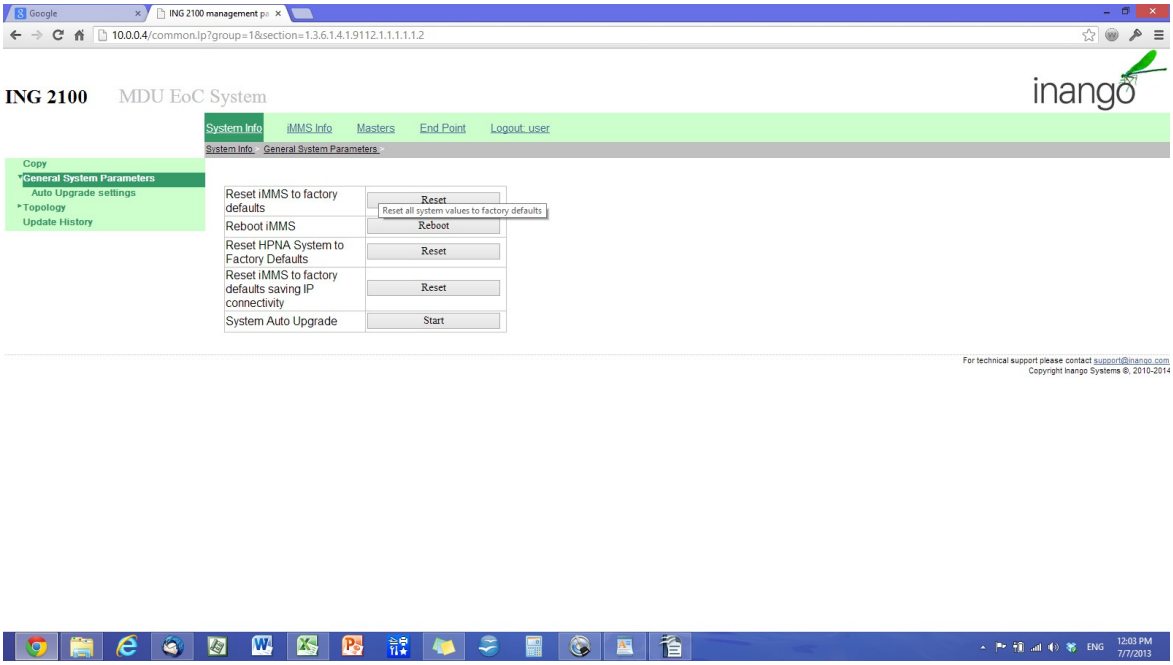


To view the status of prior “copy” operations, click on the “Update History” bar. Note that each “copy” operation is assigned an “Operation ID” (see the “Copy” screen). The same ID is used in the “Update History” screen to show the results.

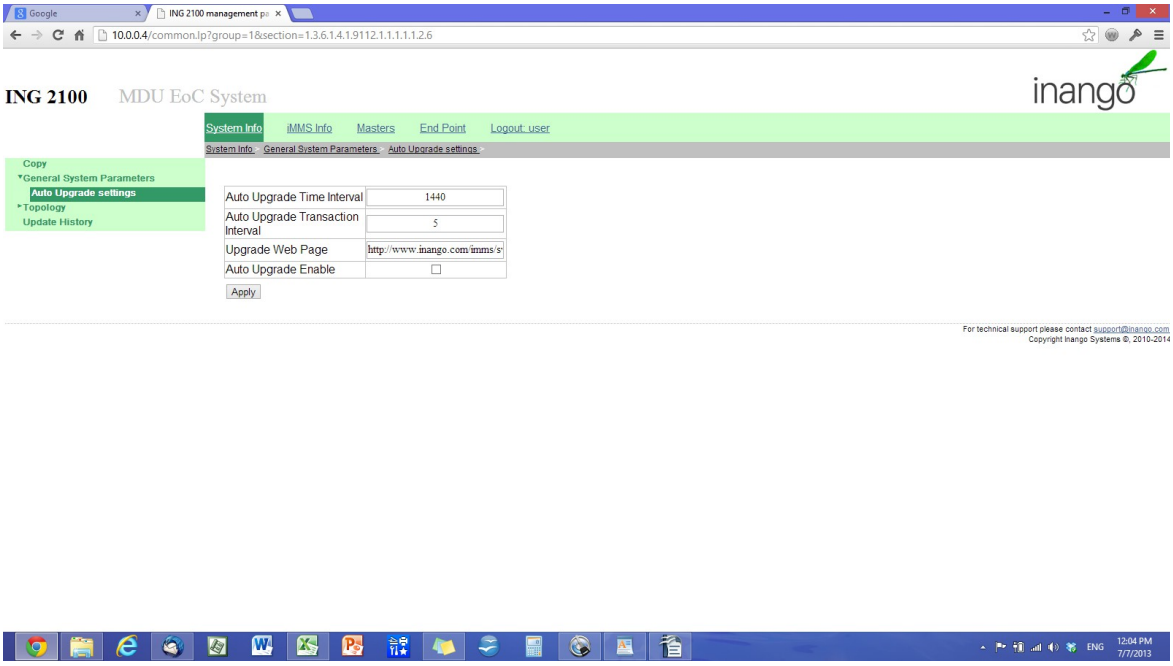


6.5.2 Generic System Parameters

This screen contains the global resets, reboot and revert to factory default options. Just click any of the shown boxes to perform the desired operation.



6.5.3 Auto-upgrade



6.5.4 Topology

The “Topology” bar contains sub-options that are used to view various levels of the HPNA network topology.

6.5.4.1 Master Topology Table

The Master Topology Table shows all MDU masters that are known in the network. Masters are identified by their MAC address on the HPNA interface; the table also shows the status of each.

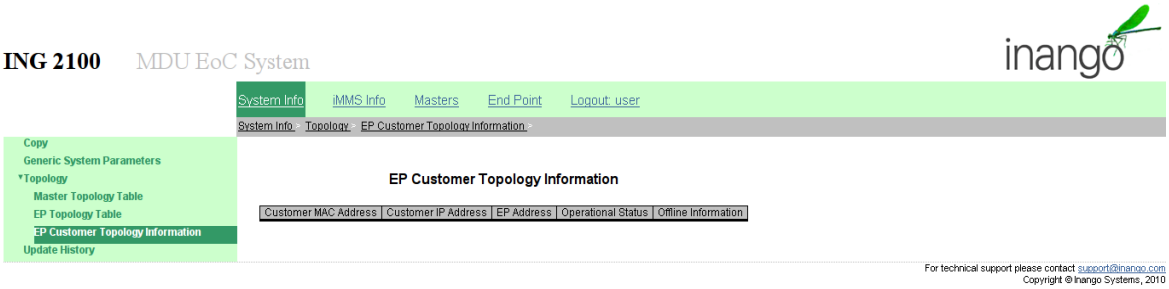
Note that a master may be “primary” or “secondary”. A master may be visible over a number of network interfaces – if so, one will be the primary ID ad all others will be classified as secondary.

6.5.4.2 EP Topology Table

This shows all endpoints “EP Topology Table” on the HPNA network. The entries are pairs: endpoint HPNA MAC address and the HPNA MAC address of the master unit to which it is connected.

6.5.4.3 EP Customer Topology Information

This screen shows the end-user equipment that is connected to each of the endpoint units on the network. These are the computers, set-top boxes, printers, etc that are connected to the endpoints in each customer's apartment.



Each row in the this table represents a network device (say a computer) in a customer's apartment. It is defined by the computer's MAC address (on the port via which is connects to the endpoint – usually it's Ethernet port), the customer's IP address (this can only be shown if DHCP Relay is enabled (see the "iMMS Info" tab for more on DHCP Relay), the endpoint (EP) HPNA MAC address (i.e. that identified the endpoint) and status information about the computer.

6.6 iMMS Info Tab

The "iMMS Info" tab is where we see and configure information related to the iMMS station. Note that this may be an embedded computer on the network or an HPNA master unit: the functions and interfaces are the same (and not related to HPNA-type operations done on the Master unit).

When you click on the "iMMS Info" tab you will see general information regarding the iMMS station, as shown below.

Licensing information is shown for reference. This is used to protect a service provider network from foreign (and unwanted) equipment. For further information, please contact Inango.

As with all other tabs, you will also see bars on the left-hand side of the screen taking you to sub-options related to the iMMS.

6.6.1 Interface Table

All network interfaces on the system running iMMS are shown. Interfaces may be of type “nic” (i.e. the iMMS traffic is sent over the network directly via a network interface card) or “vlan” (iMMS is connected via a VLAN). Other options show names (interface name and alias), status and two “enable” options.

The status information (up/down) of the interface is shown in the “Operational Status” field. If you wish to change the status, then you may do so via the “Admin Status” field.

The “enable” fields may be used as follows:

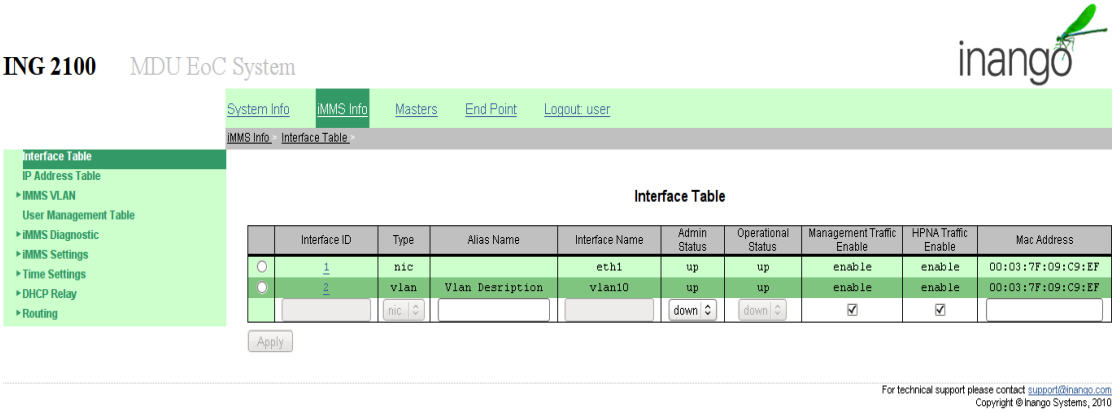
- you may set the “Management Traffic Enable” field to “disable” in order to disable all iMMS management traffic on this interface and
- you may set the “HPNA Traffic Enable” field to “disable” in order to disable HPNA traffic on this interface.

These two options will, in most circumstances, not be needed and should both be left as “enabled”. They may be used, for example, to define a management VLAN: that VLAN should be enabled for management and disabled for data and all other interfaces disabled for management and enabled for data.

6.6.1.1 Aggregated information for interfaces

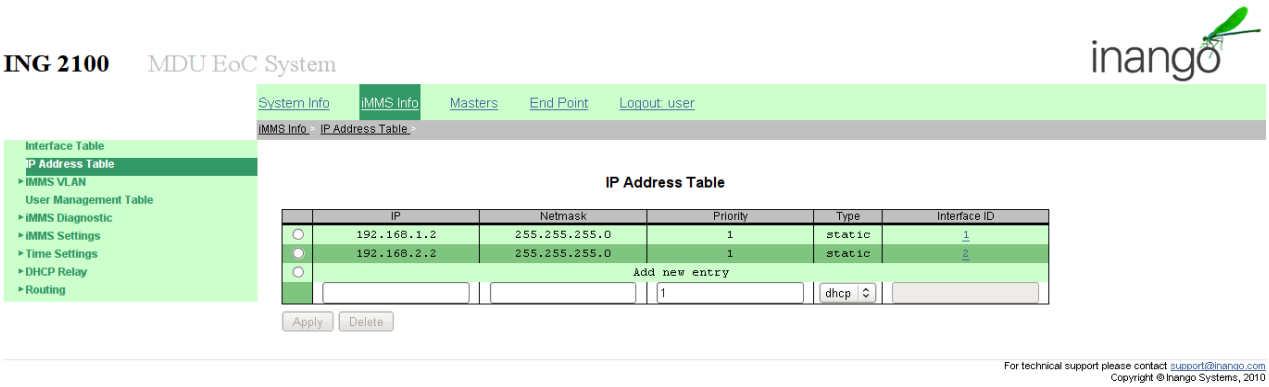
You may notice that the interface ID number is shown as a hyper-link. By clicking on this field, a pop-up screen is displayed with all information relating to this interface. The information is collected from several tables and displayed as a single table here for your convenience.

Entries may be read/written as usual. Note that when writing a field in an aggregated table, you must click on some other writeable field in order for the "Apply" button at the bottom of the table to be enabled.



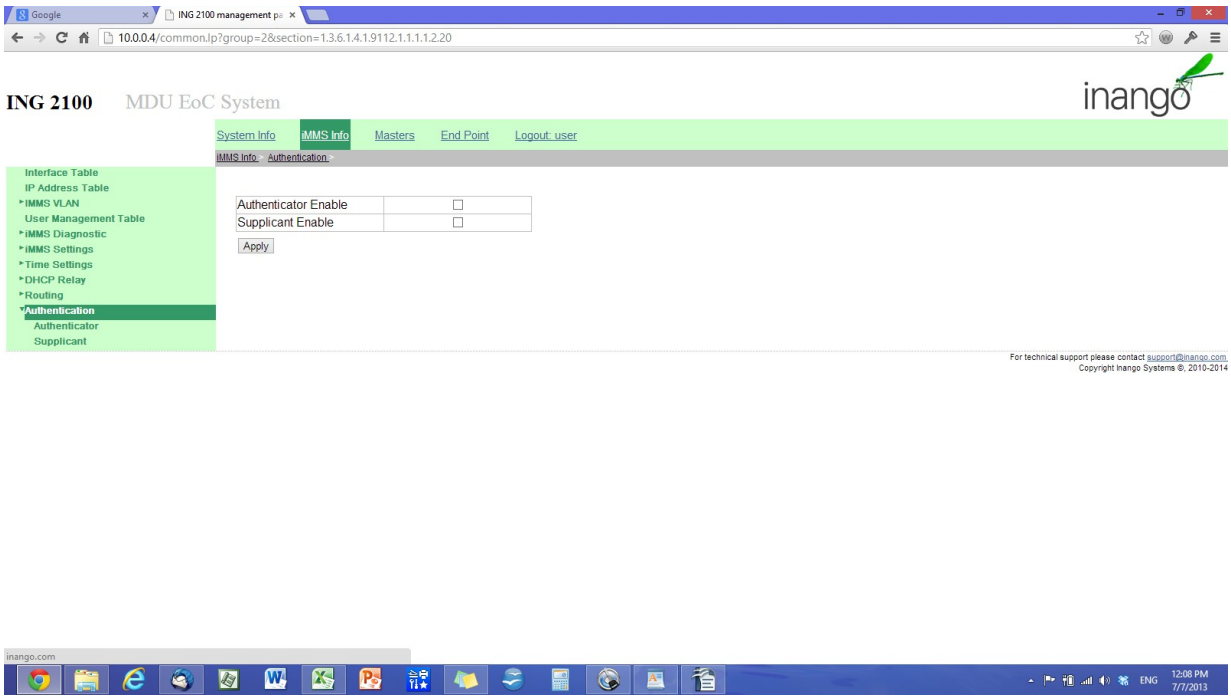
6.6.2 IP Address Table

This table shows the IP address on each network interface on the iMMS station. Note that addresses may be "static" (user-defined") or "dhcp" (if there is a DHCP Server on the network).


ING 2100
MDU EoC System
System Info
iMMS Info
Masters
End Point
Logout: user
iMMS Info
IP Address Table
Interface Table
IP Address Table
IMMS VLAN
User Management Table
iMMS Diagnostic
iMMS Settings
Time Settings
DHCP Relay
Routing
IP Address Table
IP
Netmask
Priority
Type
Interface ID
192.168.1.2
255.255.255.0
1
static
1
192.168.2.2
255.255.255.0
1
static
2
Add new entry
1
dhcp
Apply
Delete
For technical support please contact support@inango.com
Copyright © Inango Systems, 2010

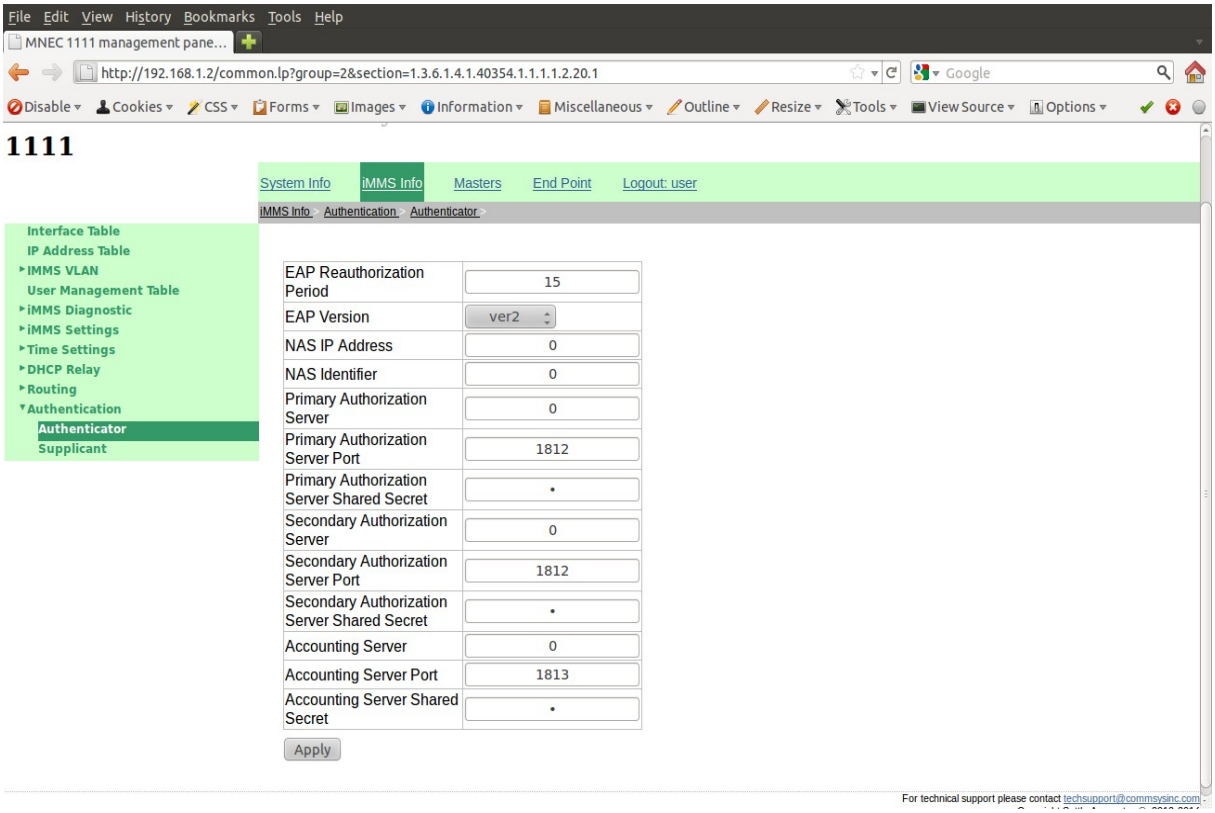
6.6.3 Authentication

This screen is used to enable Authenticator and/or Supplicant functions from the 802.1x Authentication protocol.



6.6.3.1 Authenticator

This screen contains the parameters for the 802.1x Authenticator mode.

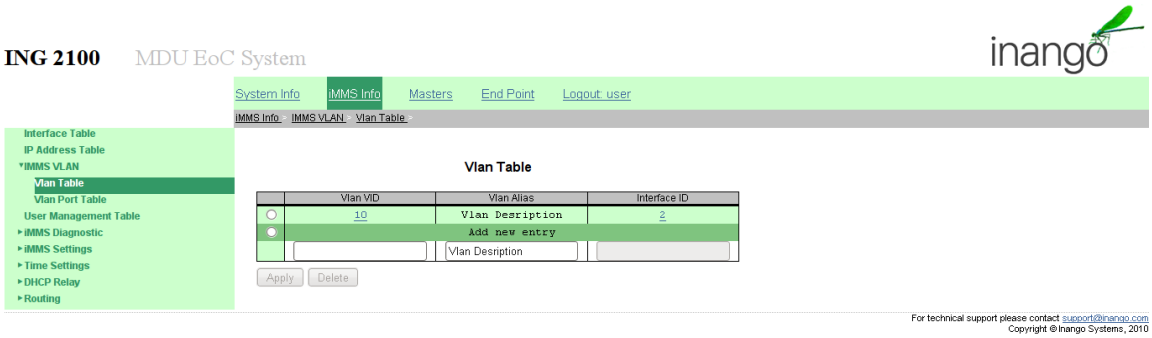


6.6.4 IMMS VLAN

This bar contains options for defining VLAN's over which the iMMS is to connect to the network. It has two sub-sections, as below.

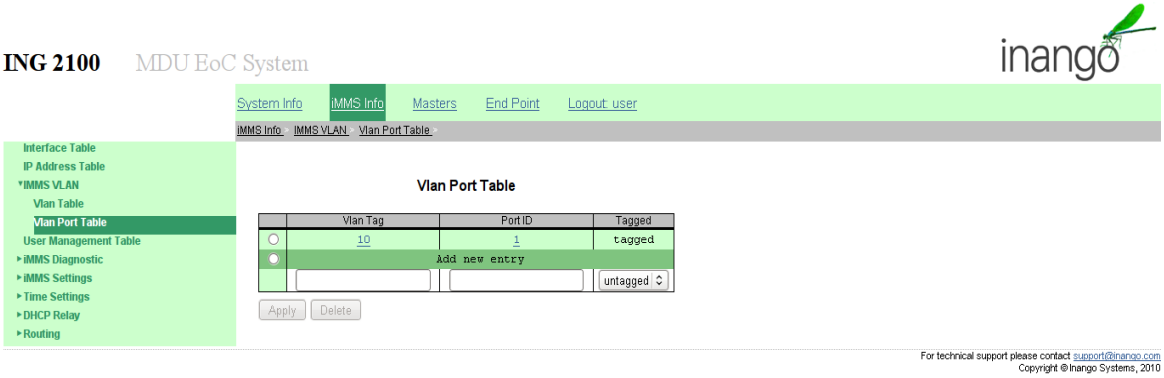
6.6.4.1 Vlan Table

This table contains the list of defined VLAN's and the ID's of their interfaces.



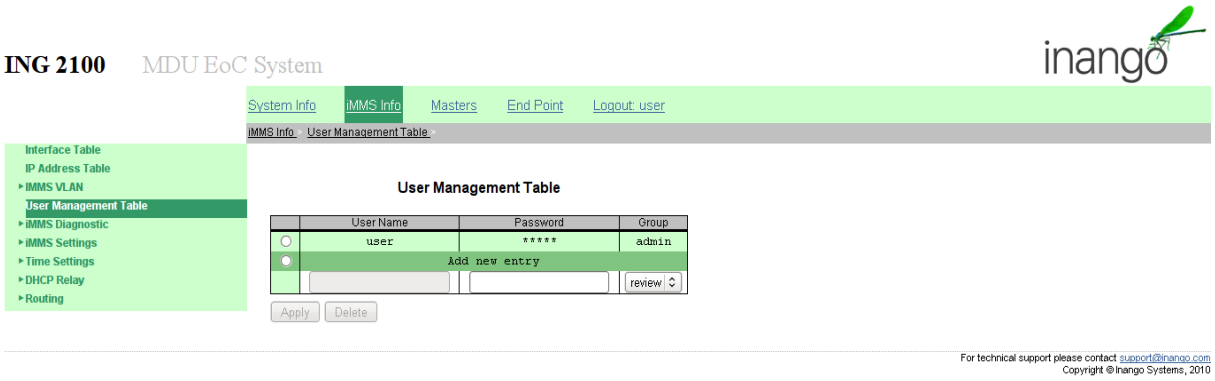
6.6.4.2 Vlan Port Table

The “Vlan Port Table” is used to define which ports belong to which (port-based) VLAN.



6.6.5 User Management Table

The User management Table is used to configure iMMS users: name, passwords and groups (or access rights). Passwords are encrypted and not visible. Only members of the “admin” group may modify entries in this table.



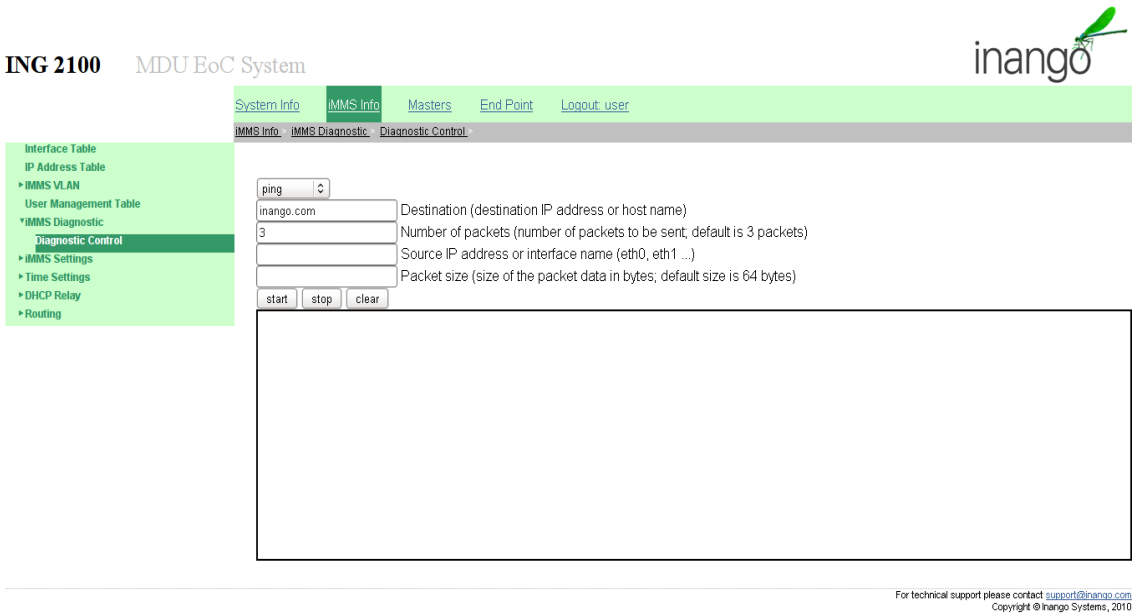
6.6.6 iMMS Diagnostics

iMMS provides a number of network diagnostics for testing connectivity of devices on the network. These are standard networking functions and not related to the HPNA Diagnostics (see the “Endpoints” tab for HPNA diagnostics).

6.6.6.1 Diagnostic Control

When you go to the “Diagnostic Control” screen, you will see that the first option in the dynamic part of the screen gives you a pull-down menu for the various supported iMMS diagnostics tests: ping, traceroute and arp.

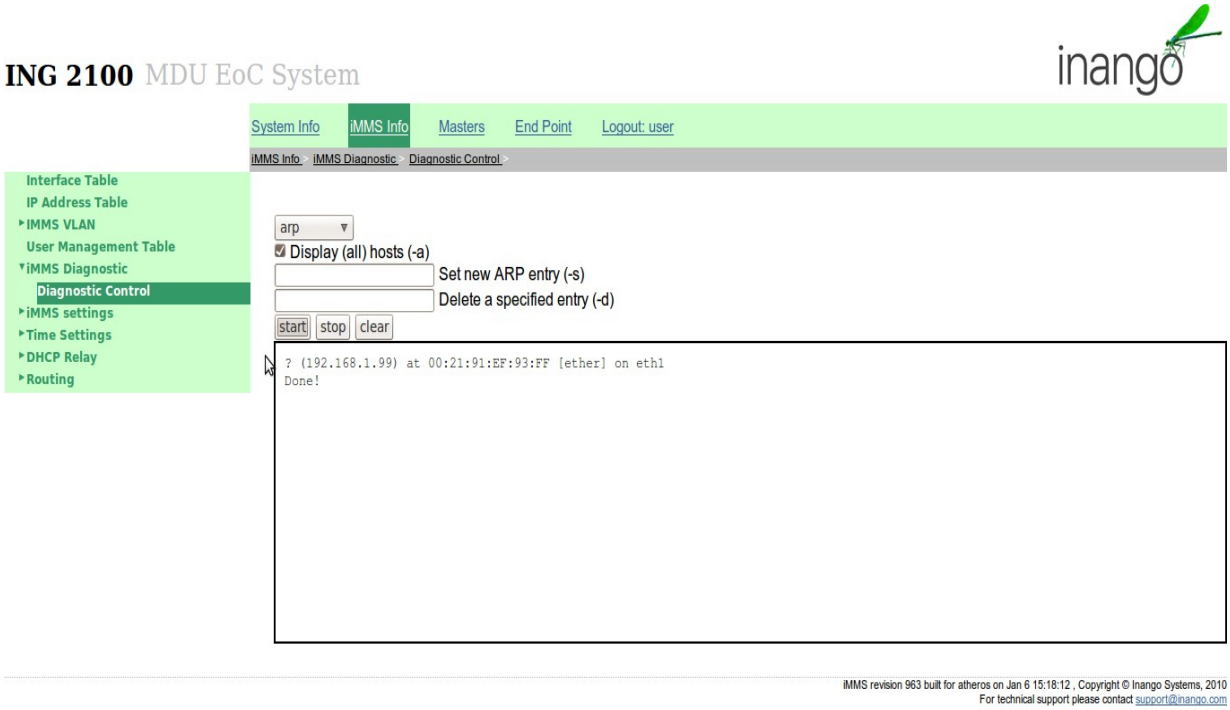
The “ping” screen looks as follows:



Enter the parameters (hostname, count and other optional parameters) and then click "start" to begin pinging the defined destination. Results are shown in the result area.

The "traceroute" just requires a hostname and then you can start the test. Results are again displayed in the result area of the screen.

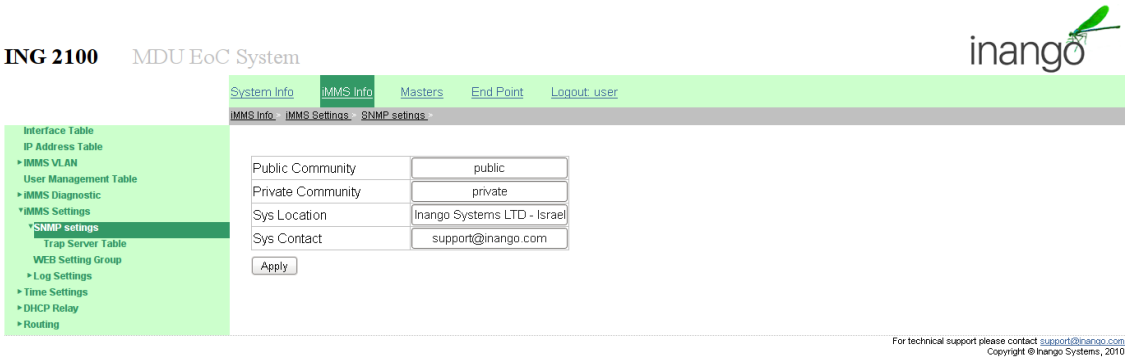
The "arp" screen is shown here:



6.6.7 iMMS Settings

The “iMMS Settings” screen lets you set global settings for different front-ends to iMMS (SNMP and web) as well as the iMMS logging functions.

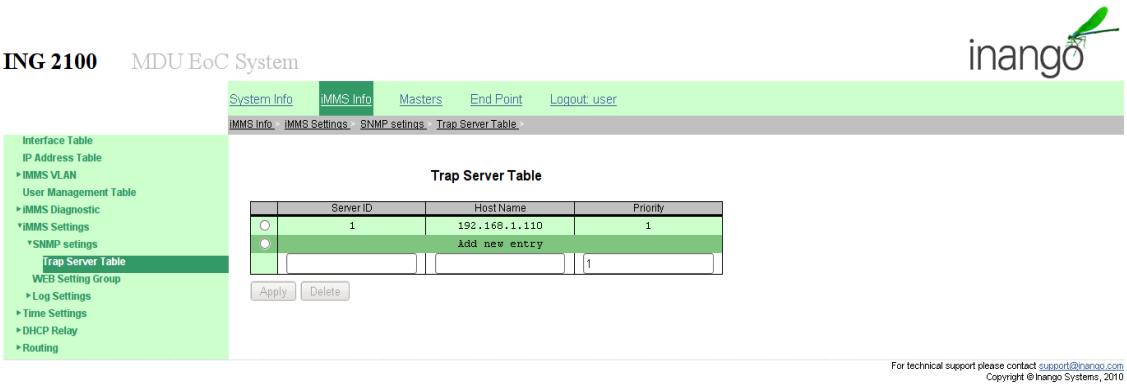
6.6.7.1 SNMP settings



SNMP-related settings are as follows:

6.6.7.1.1 Trap Server Table

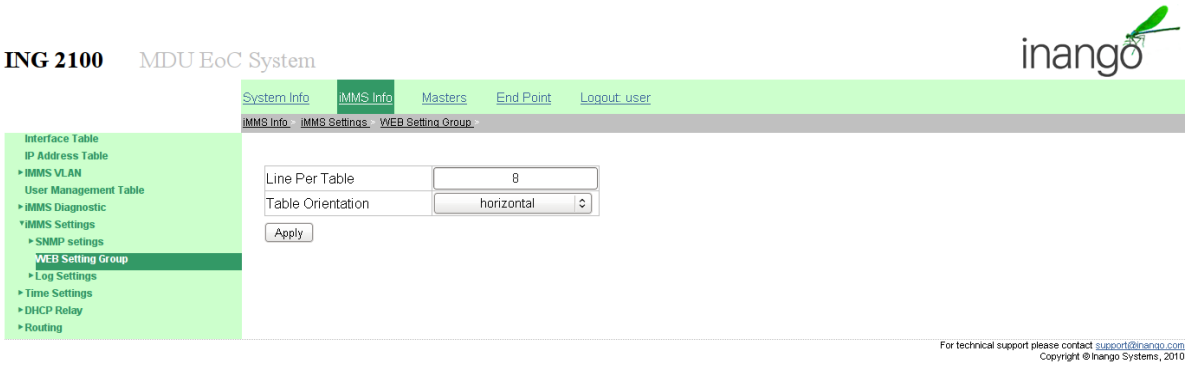
The Trap Server Table allows you to define SNMP Trap handlers.



Priorities for the TRAP Handlers define the trap handler to be used, where lower priority numbers indicate a preferred trap handler. Only if this handler is not accessible will lower priority trap handlers be chosen (i.e. those with a higher priority number).

6.6.7.2 Web Settings Group

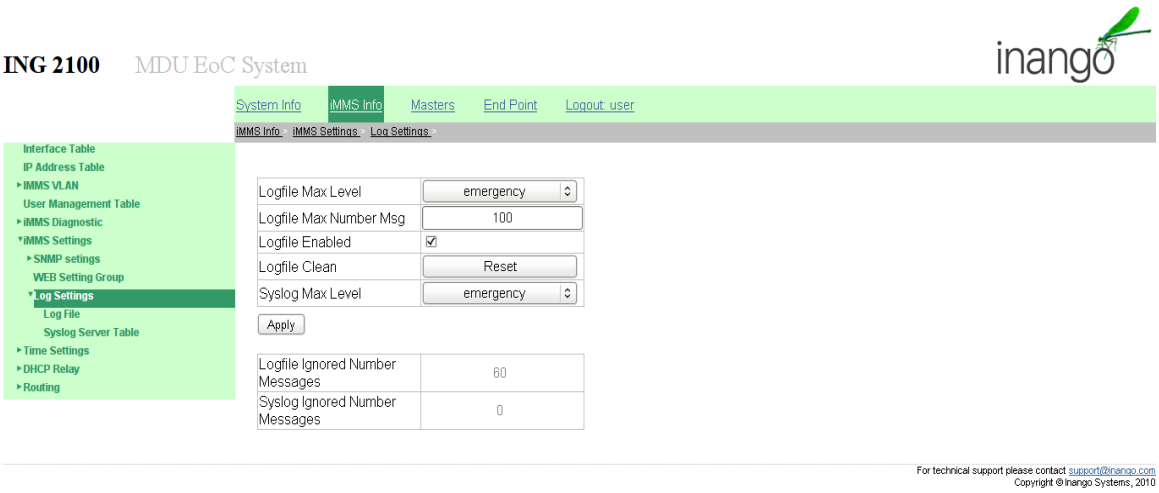
The “Web Settings Group” control options related to the web-driven user interface. These allow you to specify the number of table rows (lines) displayed on each screen and whether the fields of each table are shown horizontally or vertically. Default is 8 lines per screen and horizontal.



6.6.7.3 Log Settings

iMMS supports two types of event logging: an internal log file and a standard syslog (the latter requires a syslog server to be accessible somewhere on the network).

The “Log Setting” screen allows you to configure parameters related to the event loggers.



The Max Level defines which events are stored are logged in the internal log file, according to their severity. A pull-down option contains the possible values for this field: set the level to “emergency” (the first option in the pull-down) for minimal logging and to “debug” (the last option) for maximum event logging.

The Max Number Msg is the size of the internal log file.

Logfile Enabled must be set to enable internal logging

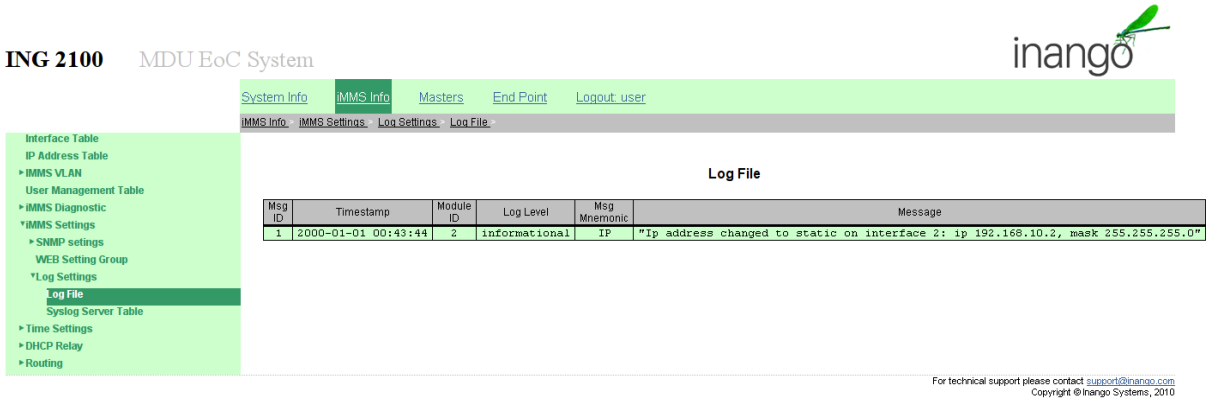
Syslog Max Level is the syslog equivalent of the Max Level field.

Click “Apply” to apply any changes you make to these fields.

Note that there are two other fields under these: they give the number of events NOT logged since the severity is not within the defined range. Separate counters are maintained for the internal log and syslog.

6.6.7.3.1 Log File

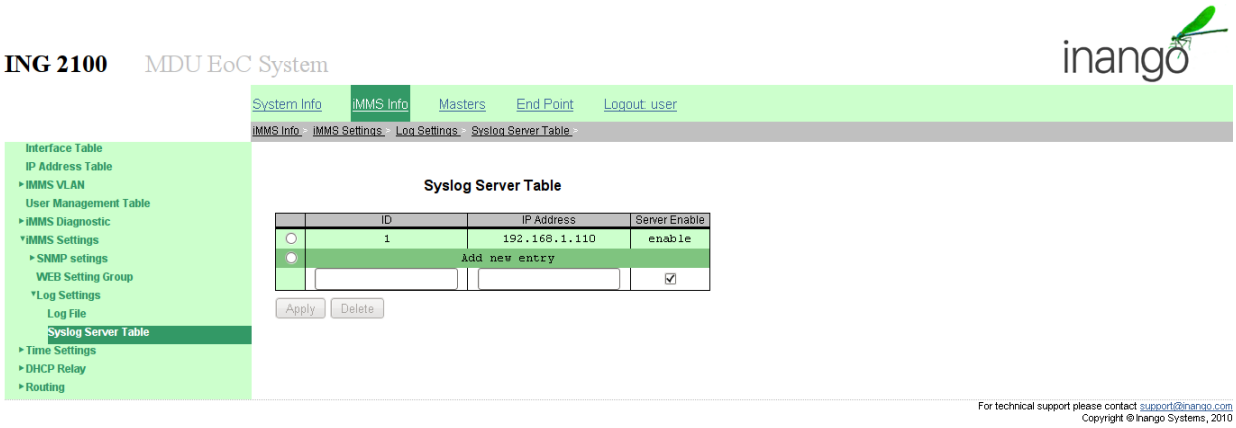
By clicking on the “Log File” bar, you may examine all events logged in the internal log file.



Some of this information is for Inango internal use only.

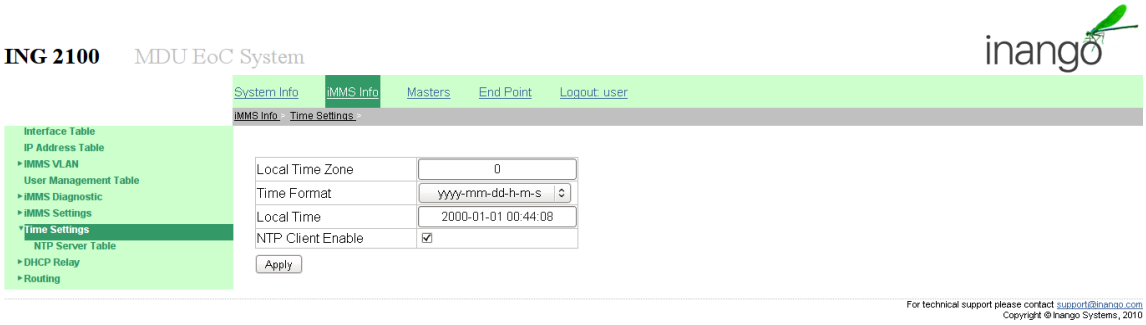
6.6.7.3.2 Syslog Server Table

This table defines the syslog servers. Each server is defined by an ID. Required information is its IP address and to check the “Enable” field.



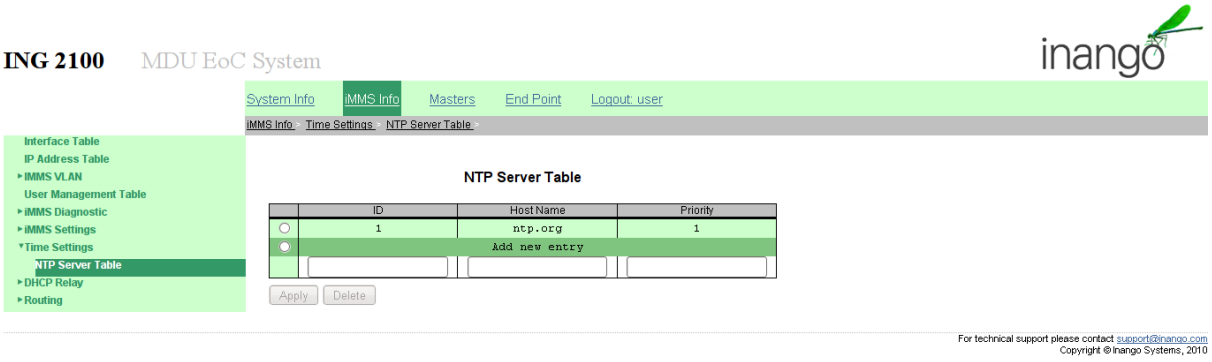
6.6.8 Time Settings

Global time-related settings.



If you selected NTP Client Enable (Network Time Protocol) then you must enter the NTP sub-screen to configure the NTP Server on the network.

6.6.8.1 NTP Server Table



The ID field defines the NTP Server. Set the IP address as appropriate. iMMS supports multiple NTP Servers and passes them all to the NTP daemon which uses them in order of designated priority.

6.6.9 DHCP Relay

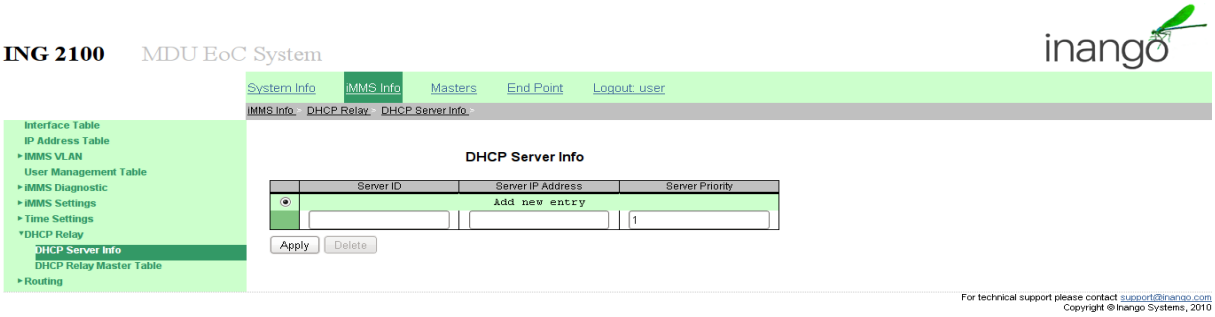
By using DHCP Relay, the iMMS can track IP addresses allocated to end-users' equipment (computers,set-top boxes, etc) in their apartments. These screens enable you to set-up and activate DHCP Relay.

In the first DHCP Relay screen, there is a single check-box that may be “clicked” in order to enable DHCP Relay. It is recommended for first visit the lower level configuration screens before checking this box.

6.6.9.1 DHCP Server Info

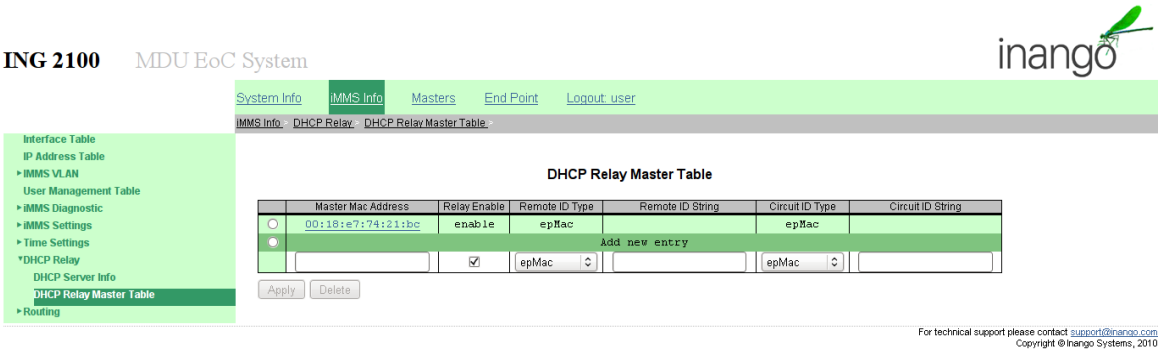
The iMMS system captures DHCP traffic and needs to forward it to a “real” DHCP Server. In order for this to happen, iMMS needs to know where the DHCP Servers reside on the network: that is defined in this table.

The “Server Priority” field is used to select between different servers if several are defined (higher numbers indicate higher priority and are chosen first if found).



6.6.9.2 DHCP Relay Master Table

This table is used to define the DHCP Relay options for each HPNA Master unit for which DHCP Relay will be enabled. These option relate to the settings of fields in the DHCP Relay control information as defined by the protocol. For full information regarding these, please refer to on page 73.

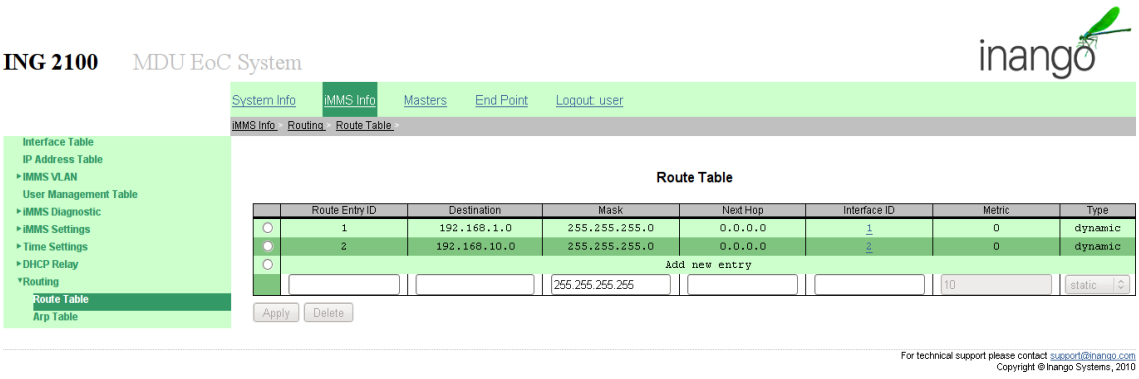


6.6.10 Routing

These screens are used to display – or configure – routing and ARP configuration for iMMS. Routing information can be inferred from the network or statically defined.

6.6.10.1Route Table

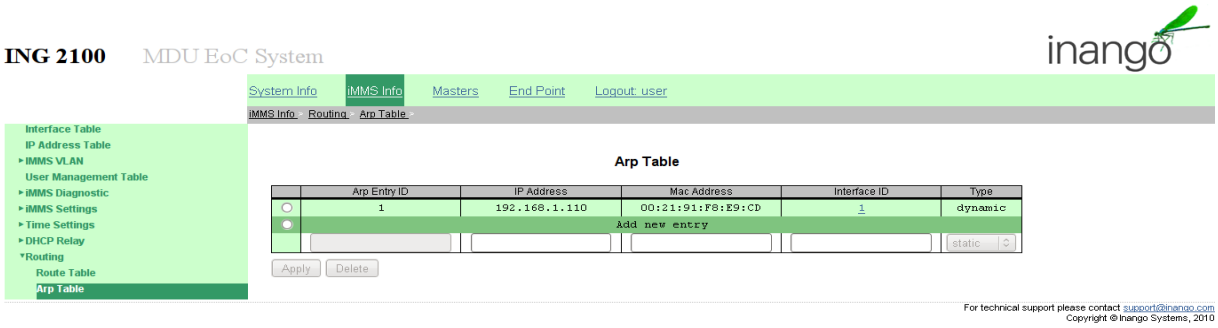
Routing information for each interface on the iMMS station:



The “Metric” field is the priority of the designated interface (a lower metric number indicates a higher priority). The “Type” field is either “dyamic” (i.e. inferred from system definition) or “static” (user-defined).

6.6.10.2ARP Table

ARP table entries in the iMMS station.



6.7 Masters Tab

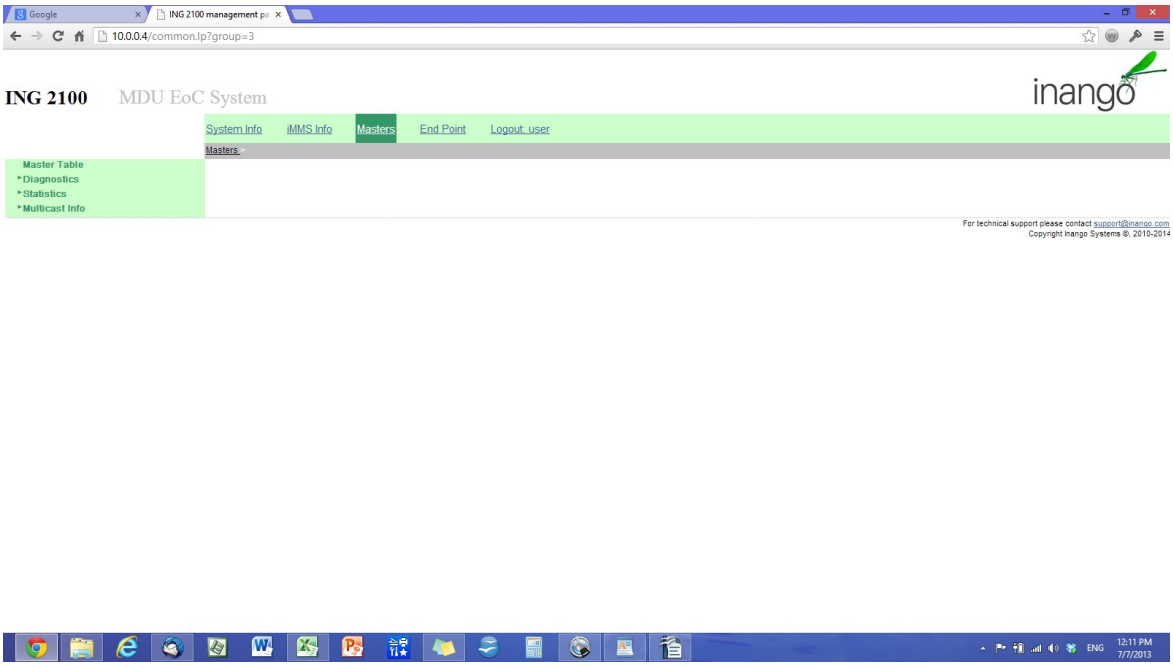
The Maters tab contains tables of information related to the MDU Masters in the network.

6.7.1 Master Table

This is the high-level per-Master table with informtion about each specific Master unit.

Most information is read-only (Manufacturer Info, Model Number, Serial Number, SW version, etc). The field “Auth” is an indication that the unit is a valid, Inango-supplied unit (it is authorized to use this HPNA network).

Note the final field in each row: “LAG”. This may be set to enable link aggregation on this Master unit. Link aggregation effectively treats two 100Mbps Ethernet interfaces as a single (200Mbps) Ethernet line and can allow you to use all of the HPNA bandwidth (usually 160Mbps) in an effective and easily managed manner. It requires the Ethernet switch feeding this master to support link aggregation (and for it to be activated).



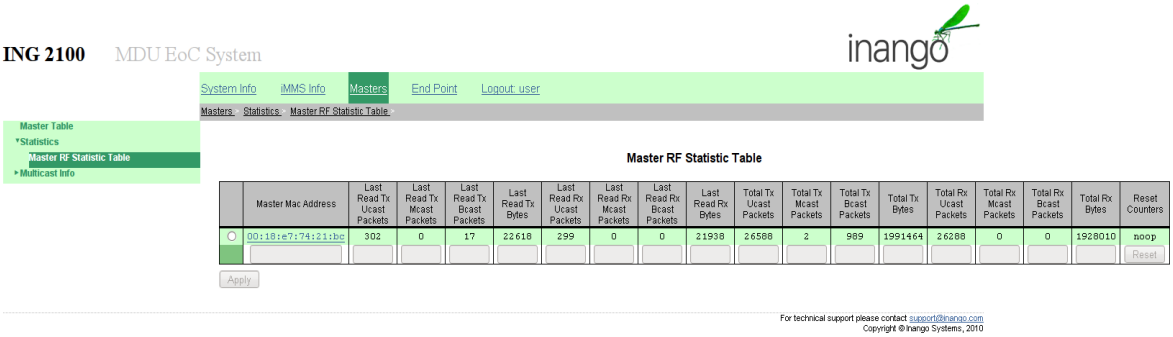
6.7.1.1 Summarized per-master information

You may notice that the MAC address of the Master units is displayed as a hyper-link. Clck on it and you will be shown a summarized (aggregated) table of information for this master. The information is gathered from multiple tables and displayed as one for your convenience.

Fields may be read/written as normal. Note that if you change any field, you must click on a different writeable field in order for the “Apply” button at the bottom of the table to be activated.

6.7.2 Statistics Table

The Statistics table contains statistics related to each master in the network.



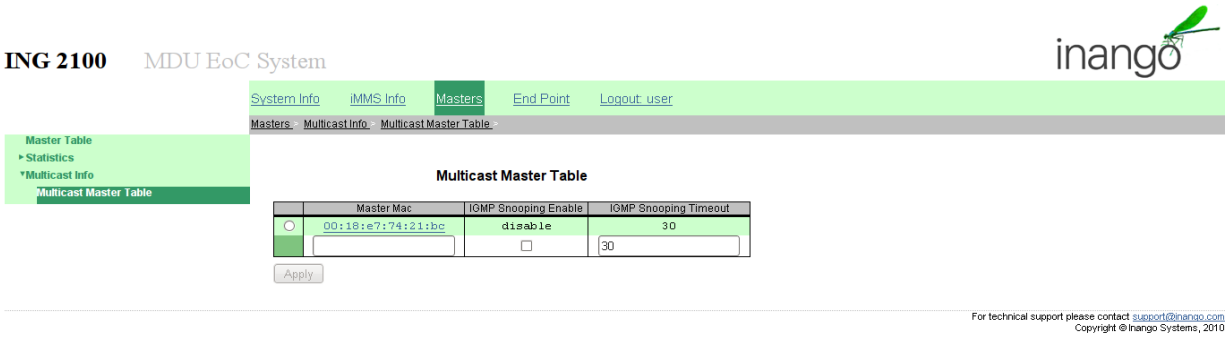
6.7.3 Multicast Info

Multicast configuration – a single sub-screen is here. Open it for configure multicast options.

6.7.3.1 Multicast Master Table

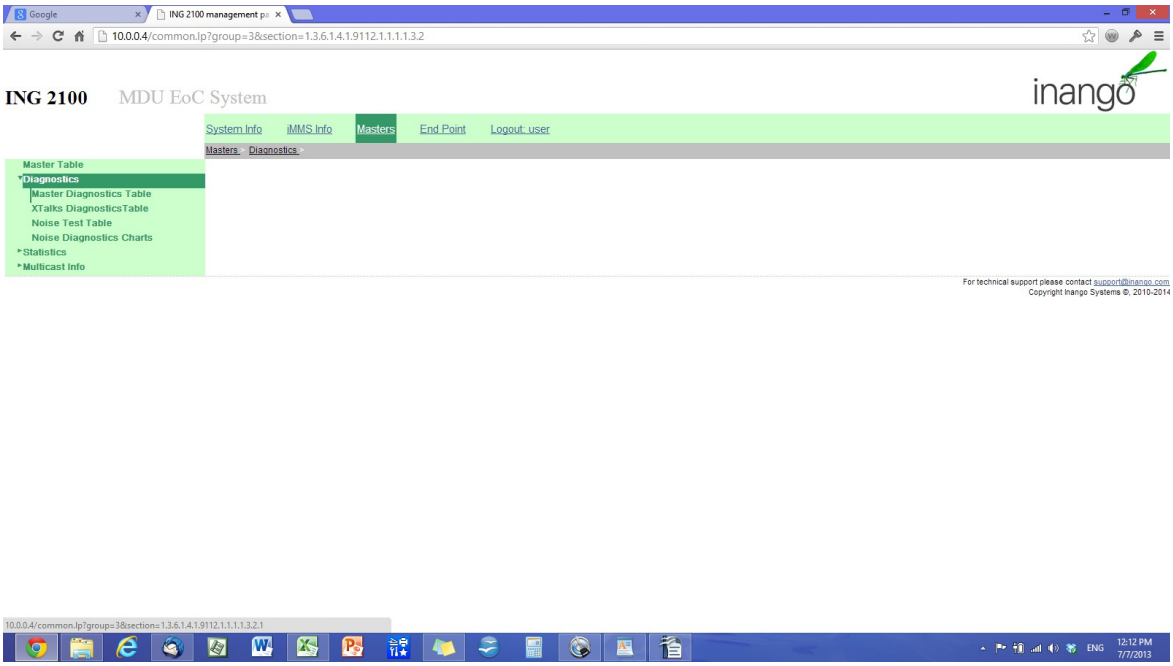
This is the multicast information for each HPNA Master.

Parameters are the IGMP Snooping time out value and a general “enable” field.



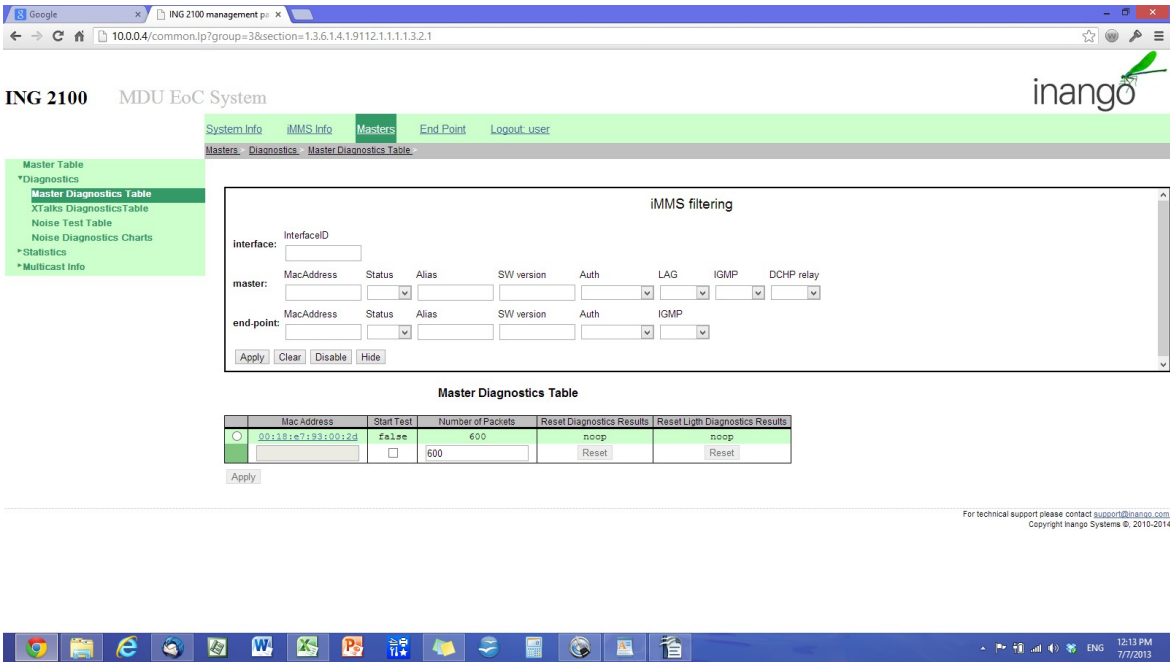
6.7.4 Master Diagnostics

As of version 10.8, when diagnostics are run on the master unit, they are performed between the master unit and all endpoints connected to it via the coaxial tree.



6.7.4.1 Master Hard diagnostics

These are the full diagnostic tests and are traffic affecting. You can use the light-weight diagnostics if traffic cannot be stopped while the tests are running.



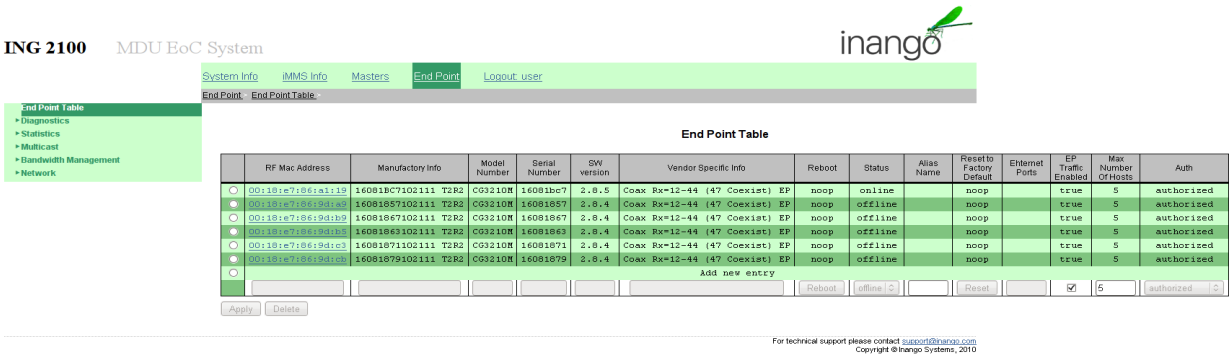
6.8 Endpoints Tab

This tab contains tables of information related to each MDU Endpoint in the network.

6.8.1 Endpoints Table

This is the high-level per-Endpoint table with informtion about each specific Endpoint unit.

Most information is read-only (Manufacturer Info, Model Number, Serial Number, SW version, etc). The field “Auth” is an indication that the unit is a valid, Inango-supplied unit (it is authorized to use this HPNA network).


ING 2100
MDU EoC System
inango
System Info
iMMS Info
Masters
End Point
Logout: user
End Point : End Point Table :
End Point Table
Diagnostics
Statistics
Multicast
Bandwidth Management
Network
End Point Table
RF Mac Address
Manufactory Info
Model Number
Serial Number
SW version
Vendor Specific Info
Reboot
Status
Alias Name
Reset to Factory Default
Ehternet Ports
EP Traffic Enabled
Max Number Of Hosts
Auth
00:10:e7:86:a1:19 16081BC71O2111 T2R2 CG3210M 16081bc7 2.8.5 Coax Rx=12-44 (47 Coexist) EP noop online noop true 5 authorized
00:10:e7:86:9d:a9 16081857102111 T2R2 CG3210M 16081857 2.8.4 Coax Rx=12-44 (47 Coexist) EP noop offline noop true 5 authorized
00:10:e7:86:9d:b9 16081867102111 T2R2 CG3210M 16081867 2.8.4 Coax Rx=12-44 (47 Coexist) EP noop offline noop true 5 authorized
00:10:e7:86:9d:b5 16081863102111 T2R2 CG3210M 16081863 2.8.4 Coax Rx=12-44 (47 Coexist) EP noop offline noop true 5 authorized
00:10:e7:86:9d:c3 16081871102111 T2R2 CG3210M 16081871 2.8.4 Coax Rx=12-44 (47 Coexist) EP noop offline noop true 5 authorized
00:10:e7:86:9d:cb 16081879102111 T2R2 CG3210M 16081879 2.8.4 Coax Rx=12-44 (47 Coexist) EP noop offline noop true 5 authorized
Add new entry
Reboot
offline
Reset
authorized
Apply
Delete
For technical support please contact support@inango.com
Copyright © Inango Systems, 2010

6.8.1.1 Summarized per-endpoint information

You may notice that the MAC address of the Endpoint units is displayed as a hyper-link. Clck on it and you will be shown a summarized (aggregated) table of information for this endpoint. The information is gathered from multiple tables and displayed as one for your convenience.

Fields may be read/written as normal. Note that if you change any field, you must click on a different writeable field in order for the “Apply” button at the bottom of the table to be activated.

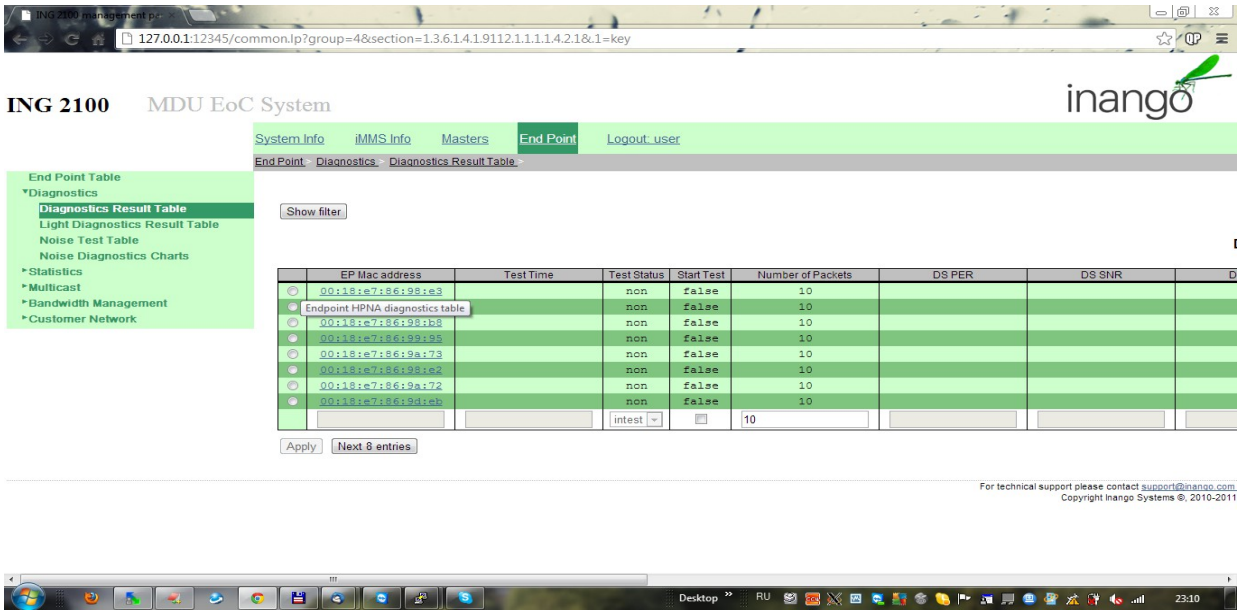
6.8.2 Diagnostics

HPNA Diagnostics tests may be run to gather information about line quality: performance, noise levels, power, etc. These tests are activated and results read through tables in this group.

Two sub-sections are provided for HPNA diagnostics: full diagnostics (which are traffic affecting) and light-weight diagnostics (which do not affect HPNA traffic while running).

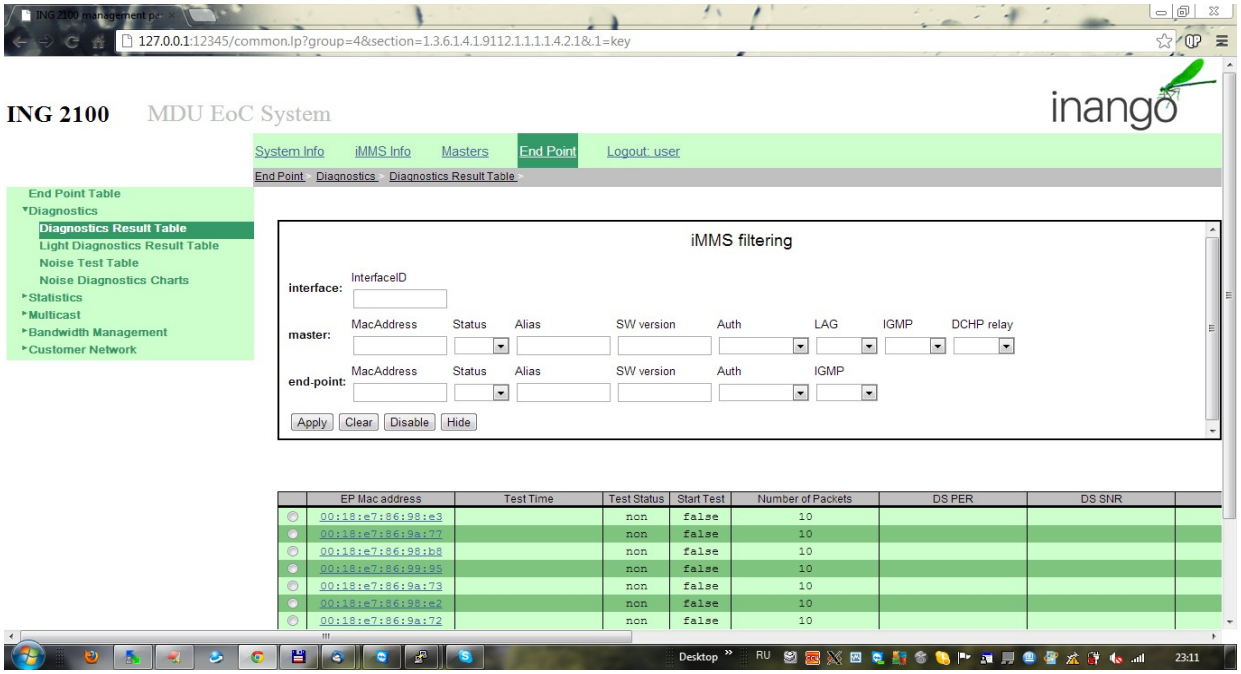
The full tests generate information on PER (packet error rate), SNR (signal-noise ratio), data rate, power and PHY rate – all in both upstream and downstream directions. The only difference between these and the lightweight diagnostics is that the latter do not include PER information.

Tests are configured by setting a number of packets (default is 10), set “Start Test” to “true” and then “Apply” to run the test. The table can be re-read until “Status” shows that the tests have completed – at this time the results are displayed in the appropriate fields of this row.



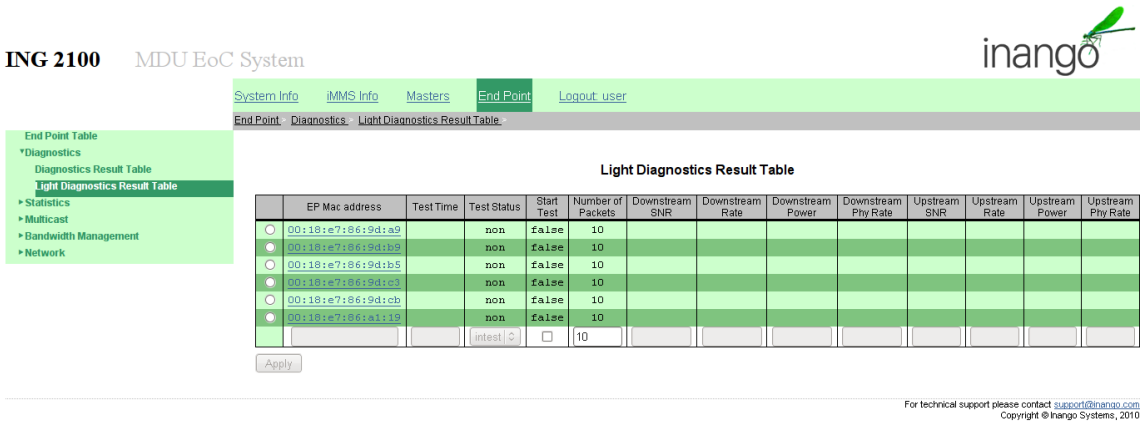
6.8.2.1.1 Diagnostics Result Table

The format of the full diagnostics test table is as follows:



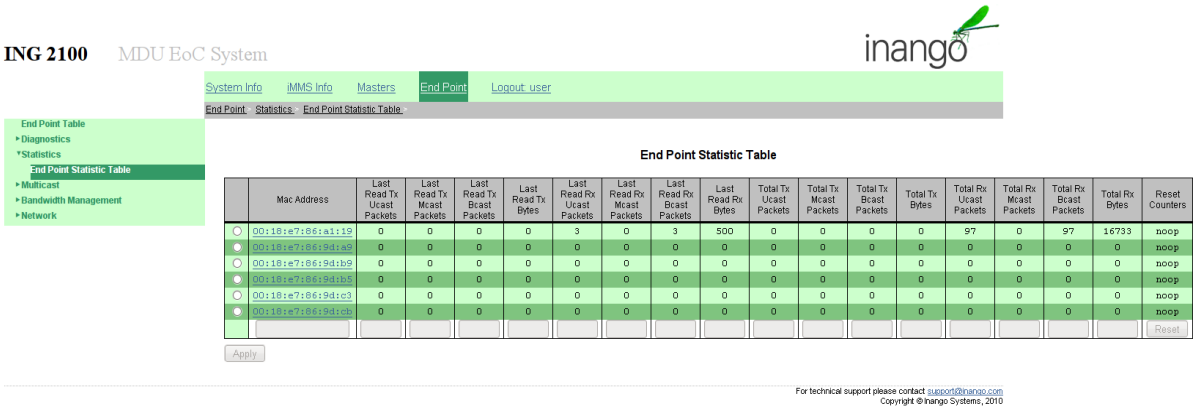
6.8.2.2 Light Diagnostics Results Table

The format of the lightweight diagnostics tests table is almost identical – just no PER values.



6.8.3 Endpoint Statistic Table

Statistics related to each endpoint's activity.



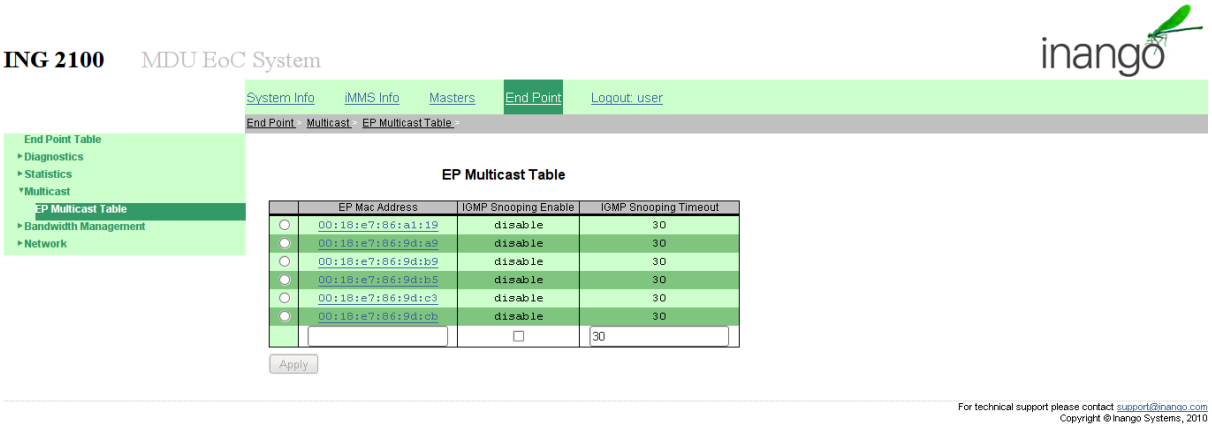
6.8.4 Multicast

Multicast configuration – a single sub-screen is here. Open it for configure multicast options.

6.8.4.1 EP Multicast Table

This is the multicast information for each HPNA Endpoint.

Parameters are the IGMP Snooping time out value and a general “enable” field.



6.8.5 Bandwidth Management

Upstream and downstream bandwidth can be controlled for each HPNA endpoint.

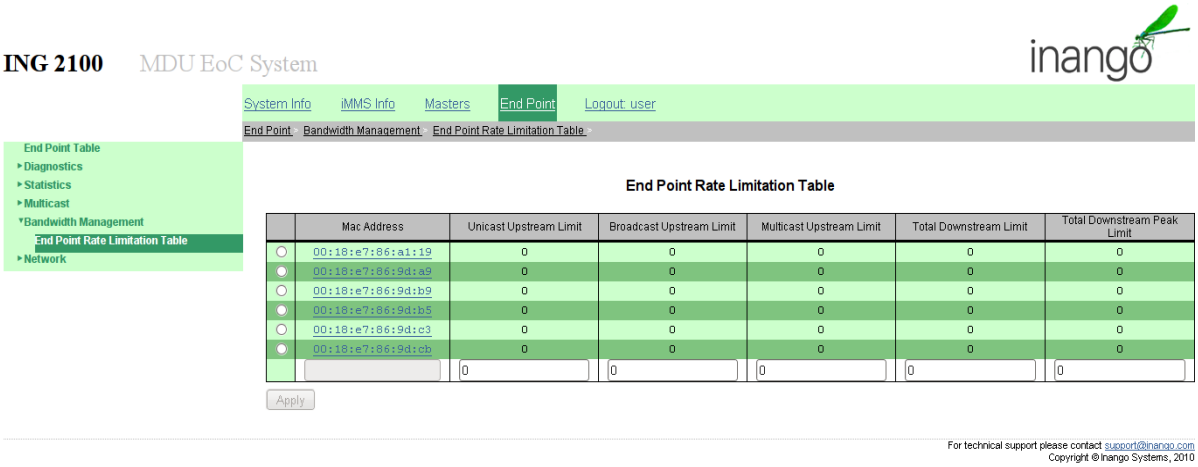
Of these, the more useful is usually the upstream limitation. If there is 160Mbps of bandwidth on the HPNA network, each endpoint can at any given point in time transmit at the full line rate (160Mbps). If this is not appropriate for your application (e.g. you wish to guarantee that some bandwidth is always available for some other stations' use) then you can restrict the upstream (HPNA) bandwidth at which this endpoint will be allowed to transmit.

Downstream rate limitation involves buffering data received from the HPNA network and transmitted on the Ethernet ports to keep the traffic to within the defined rate.

This configuration is done through the Endpoint rate Limitation Table.

6.8.5.1 Endpoint Rate Limitation Table

The format of the endpoint rate limitation table is shown below. Note that upstream (Eth-to-HPNA) traffic may be limited according to frame type (unicast, multicast or broadcast) whereas downstream traffic (HPNA-to-Eth) is handle globally.



6.8.6 Network

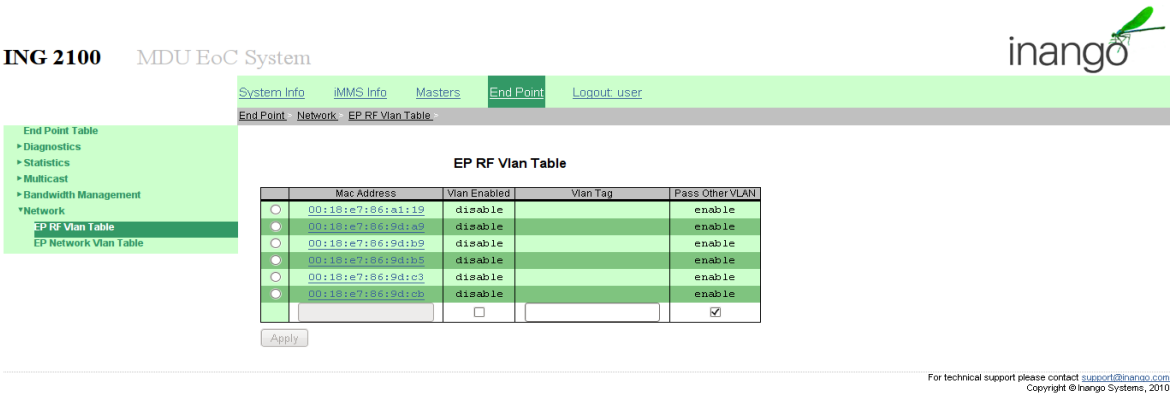
Network options are currently all related to VLAN's and their configuration.

iIMMS supports two types of VLAN's: those defined on the HPNA interface and VLAN's defined on the network (Ethernet) interfaces of each endpoint.

6.8.6.1 EP RF Vlan Table

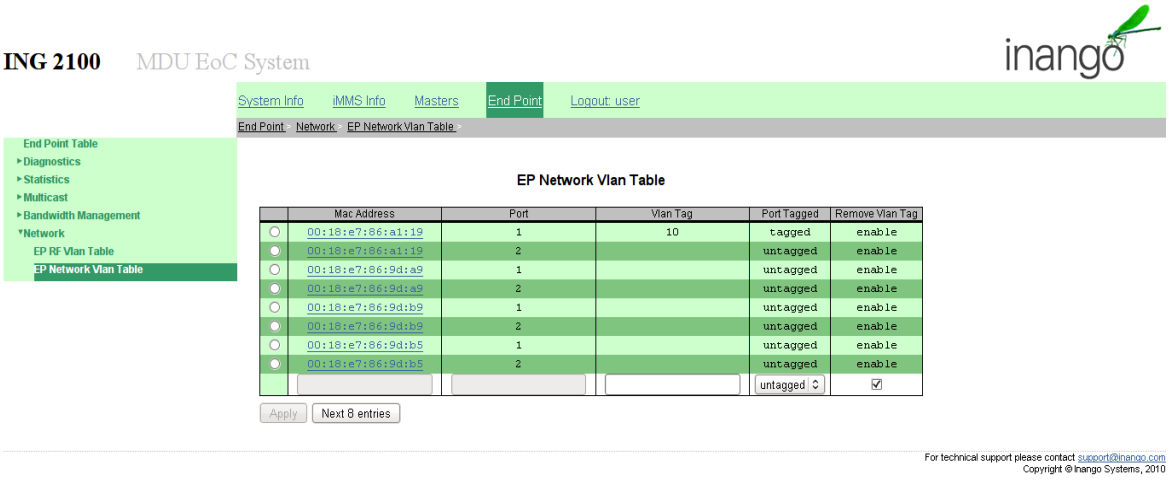
This is the table used to define VLAN handling on the HPNA interface of an Endpoint unit.

If VLAN's are enabled (click the “Vlan Enabeld” field) then we can define the tag to be added (the “Vlan tag” field) and whether other VLAN's are to be passed through or not (“Pass Other Vlans”).



6.8.6.2 EP Network Vlan Table

This table defines VLAN handling on each of the ports (Ethernet interfaces) of each Endpoint.



7.

This chapter provides more details on the some of the more complex commands and operations (i.e. commands with multiple and not always self-explanatory parameters) in iMMS.

7.1 Copy

The Copy operation is used to backup and restore configuration data bases, software (or firmware) upgrades and in installing new license files. The source or destination of the copy operation can be any of the following devices: a computer, the iMMS station, an HPNA Master unit or an HPNA Endpoint unit.

Parameters and use-case examples for the “copy” operation follow below.

Calling Syntax	`copy <source-url> <dest-url>`
Parameters	
`tftp://{ip-address}/<filename>` Copy via TFTP to/from the named file at the designated IP address	
`[local://]<fileType>`	
`master://{macAddress }/| all<fileType>` Copy HPNA firmware to Master unit at the designated HPNA MAC address (or all masters)	
`ep://{macAddress	all}/<fileType>` Copy HPNA firmware to Endpoint unit at the designated HPNA MAC address (or all EP's)	
File Types	
`firmware|image|license|dbrunningconfig`	
Valid values for source-url	
`dbrunningconfig`	
`tftp://<ip-address>/<filename>`	
Valid values for dest-url	

dbrunningconfig
image
kernel
rootfs
license
master://<mac-address>/firmware
ep://<mac-address>/firmware
tftp://<ipaddress>/filename
Examples
Upgrade new iMMS software image from file called new.image on station 10.0.0.10 `copy tftp://10.0.0.10/new.image local://image`
Install new firmware on an endpoint at MAC address 11:22:33:44:55:66 `copy tftp://10.0.0.10/new.ep.fw ep://11:22:33:44:55:66/firmware`
Export current data base configuration to file on server `copy local://dbrunningconfig tftp://10.0.0.10/saved.db`
Update iMMS license file `copy tftp://10.0.0.10/new.license local://license`

7.2 Ping

iMMS can send regular “ping”s to test connectivity of devices on the network.

Calling Syntax	`ping <destination> [size {packet_size}] [count {packet_count}] [timeout {time_out}]`
Parameters	
destination	Destination IP address to ping
size	Size (in bytes) of packets to send (default is 56)
count	Number of packets to send 1 – 65535 (default is 4)

timeout	Time to wait for response in millisecs 50 – 65535 (default is 200)
Examples	
`Ping 10.0.0.10 size 1000 count 10 timeout 20`	

7.3 Traceroute

The “tracetroute” operation can be used to find the actual path that IP packets will take to their destination.

Calling Syntax	`traceroute <destination>`
	`[ttl max-ttl [count packet_count]`
	`[timeout time_out [tos tos] [packet_size]`

Parameters	
destination	Destination IP address
ttl	Time-to-live in terms of number of hops (default is 30)
count	Number of probes to be sent to each TTL level (default is 3)
timeout	Number of seconds to wait for response to a probe (default is 3)
tos	Type-Of-Service byte in the packets' IP header
packet_size	Number of bytes per packet (default is 40)
Examples	
`traceroute 10.0.0.10 ttl 25 count 5 timeout 3 64`	

7.4 iMMS Configuration Commands

The following commands configure various aspects of the iMMS station.

7.4.1 IP Configuration

Define an IP address for an interface.

Calling Syntax	`hpna mngm ip config <interface> [address <value>] [mask <value>] [prio <value>] [type <value>]`
Parameters	
`interface`	Interface ID (from interface table)
`address`	IP address
`mask`	Network mask: in format 255.255.255.0
`prio`	Priority metric of this entry (lower number = higher priority)
`type`	Static (1) or dhcp (2)
Examples	
`hpna mngm ip config 1 address 192.168.1.1 mask 255.255.255.0 type 1`	

7.4.2 VLAN Configuration

7.4.2.1 VLAN Port configuration

Add a port to a VLAN and define whether tagging is to be done or not.

Calling Syntax	`hpna mngm vlan vlan-port config <vlan-tag> <port-ifindex> [enable-tag <value>]`
Parameters	
`vlan-tag`	VLAN-id
`port-ifindex`	Port number in interface table
`enable-tag`	Set to either TRUE (1) or FALSE (2)
Examples	
`hpna mngm vlan vlan-port config 7 3 1`	

7.4.2.2 VLAN Interface

Define VLAN's in the system and optionally define an alias (name) for them.

Calling Syntax	`hpna mngm vlan vlan-if config <vid>` `[alias <value>]`
Parameters	
`vid`	VLAN-id (tag)
`alias`	Alias name for this VLAN interface
Examples	
`hpna mngm vlan vlan-if config 7 "home"`	

7.4.3 DHCP Relay

This command configures an HPNA master that is to handle DHCP Relay. Note that DHCP Relay must first be enabled globally at the iMMS level.

Calling Syntax	`hpna mngm dhcp-relay master config <master-mac>`
	`[enable-relay <value>] [remote-type <value>]`
	`[remote-string <value>]`
	`[circuit-type <value>]`
	`[circuit-string <value>]`

Parameters	
`master-mac`	MAC address of HPNA master on which DHCP Relay is to be enabled/disabled
`enable-relay`	Boolean: set to TRUE (1) or FALSE (2) to enable/disable DHCP Relay on this port
`remote-type`	This defines the type of "remote string" to be used in DHCP Relay packets: 1: Endpoint HPNA MAC address (this is the default value) 2: Master HPNA MAC address 3: User-defined (in remote-string)
`remote-string`	See above

circuit-type	This defines the type of “circuit string” to be used in DHCP Relay packets: 1: Endpoint HPNA MAC address 2: Master HPNA MAC address (this is the default value) 3: User-defined (in remote-string)
circuit-string	See above
Examples	
`hpna mngm dhcp-relay master config 11:22:33:44:55:66 1 rt rs ct cs`	

7.4.4 Routing Table

Update routing information on the iMMS station

Calling Syntax	`hpna mngm route rt config <entry-id> [dest <value>] [mask <value>] [next-hop <value>] [if <value>] [metric <value>]`
Parameters	
entry-id	ID of this entry
dest	Destination IP address
mask	Mask for destination IP
next-hop	Next hop IP address
if	Interface ID of next hop
metric	Priority metric for this route
Examples	
`hpna mngm route rt config 1 dest 2.58.7.16 mask 255.255.255.0 next-hop 4.68.2.2 if 3`	

7.4.5 ARP Table

Add or update ARP table information on the iMMS station.

Calling Syntax	`hpna mngm route arp config <entry-id> [dest <value>] [mac <value>] [if <value>]`
Parameters	
`entry-id`	ID of entry to be added (or updated)
`dest`	Destination IP address
`mac`	Destination MAC address
`if`	Interface ID for destination
Examples	
`hpna mngm route arp 1 dest 10.0.0.1 mac 66:55:44:33:22:11 if 2`	

7.5 HPNA Master Commands

7.5.1 Configure HPNA Master

This command is used to configure HPNA Master units.

Calling Syntax	`hpna master info config <master-mac> [reboot <value>] [reset <value>] [reset-subsystem <value>] [lag <value>]`
Parameters	
`master-mac`	HPNA MAC address of Master unit to be configured
`reboot`	Boolean: set to TRUE (1) to reboot or FALSE (2) to not
`reset`	Boolean: set to TRUE (1) to reset unit or FALSE (2) to not
`reset-subsystem`	Boolean: set to TRUE (1) to reset the HPNA sub-system or FALSE (2) to not

lag	Boolean: set to TRUE (1) to enable Link Aggregation or FALSE (2) to disable LAG
Examples	
`hpna master info config 11:22:33:44:55:66 reboot 2 reset 2 reset-subsystem 2 1`	

7.5.2 Multicast Configuration

Configure multi-cast operations on the HPNA Master unit.

Calling Syntax	`hpna master multicast master-mcast config <master-mac> [igmp <value>] [igmp-timeout <value>]`
Parameters	
master-mac	HPNA MAC address of Master unit
igmp	Boolean: enable (1) or disable (2) IGMP snooping
igmp-timeout	IGMP timeout value in seconds (default is 30)
Examples	
`hpna master multicast master-mcast config master-mac 11:22:33:44:55:66 igmp 1`	

7.6 HPNA Endpoint Commands

7.6.1 Configure HPNA Endpoint

This command is used to configure HPNA Endpoint units.

Calling Syntax	`hpna ep info config <ep-mac> [reboot <value>] [alias-name <value>] [reset <value>]`

	`[enable-traffic <value>]` `[host-number <value>]`
Parameters	
`ep-mac`	Endpoint HPNA MAC address
`reboot`	Boolean: set to TRUE (1) to reboot or FALSE (2) to not
`alias-name`	Alias name (description) of this endpoint
`reset`	Boolean: set to TRUE (1) to reset the unit or FALSE (2) to not
`enable-traffic`	Boolean: set to TRUE (1) to enable data traffic on this unit or FALSE (2) to disable traffic
`host-number`	Maximum number of peripheral units that may be connected to this endpoint
Examples	
`hpna ep info config 11:22:33:44:55:66 alias-name jjj enable-traffic 1 host-number 4`	

8. MIB

The entire MIB for iMMS is shown below:

```
                INANGO-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       -- mib-2,
       TimeTicks,
       Counter32,
       Counter64,
       -- UInteger32,
       Integer32,
       IpAddress,
       enterprises
           FROM SNMPv2-SMI

       TEXTUAL-CONVENTION,
       MacAddress,
       DisplayString,
       -- TimeStamp,
       TruthValue,
       RowStatus
           FROM SNMPv2-TC
       ;

EnableEnum ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "Represents a enable value."
    SYNTAX       INTEGER { enable(1),disable(2) }

RebootEnum ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "Represents a reboot value."
    SYNTAX       INTEGER { reboot(1),noop(2) }

ResetEnum ::= TEXTUAL-CONVENTION
```

```
    STATUS current
    DESCRIPTION
            "Represents a reboot value."
    SYNTAX       INTEGER { reset(1),noop(2) }


StartEnum ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "Represents a start value."
    SYNTAX       INTEGER { start(1),noop(2) }


       -- common configuration for all devices

       inango  MODULE-IDENTITY
         LAST-UPDATED "201307021825Z"
         ORGANIZATION "Inango Systems LTD."
         CONTACT-INFO "contact@inango.com"
         DESCRIPTION
                "The structure of management information for the HPNA
3.1 MDU management."
         ::= { enterprises 9112 }


products OBJECT IDENTIFIER ::={ inango  1 }



hpna OBJECT IDENTIFIER ::={ products  1 }



hpna31MDU OBJECT IDENTIFIER ::={ hpna  1 }



hpna31MDUproducts OBJECT IDENTIFIER ::={ hpna31MDU  2 }
```

```
hpna31MDUdeviceMib OBJECT IDENTIFIER ::={ hpna31MDU  1 }



hpna31MDUnotification OBJECT IDENTIFIER ::={ hpna31MDU  3 }



systemGroup OBJECT IDENTIFIER ::={ hpna31MDUdeviceMib  1 }



copyTable OBJECT-TYPE
SYNTAX SEQUENCE OF CopyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Copy (download or upload) software, firmware or configuration
information"
::={ systemGroup  1 }




copyEntry OBJECT-TYPE
SYNTAX  CopyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in copyTable - shows results of past copy operations"
INDEX {

          operationId
 }
::={ copyTable  1 }

CopyEntry::= SEQUENCE {
```

```
          operationId    Integer32 ,
          sourceProtocol    INTEGER ,
          sourceType    INTEGER ,
          sourceName    DisplayString ,
          sourceFileName    DisplayString ,
          sourceFileType    INTEGER ,
          destProtocol    INTEGER ,
          destType    INTEGER ,
          destName    DisplayString ,
          destFileName    DisplayString ,
          destFileType    INTEGER ,
          bytesRead    Integer32 ,
          bytesWrite    Integer32 ,
          operationStatus    INTEGER ,
          abort    TruthValue ,
          start    TruthValue ,
          copyTableRowStatus    RowStatus
 }
```

```
operationId OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Operation counter for (internal use)"
::={ copyEntry  1 }
```

```
sourceProtocol OBJECT-TYPE
SYNTAX INTEGER   {
      tftp(3),local(5),http(6),https(7)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Source file copy protocol (ftp, tftp, scp)"
::={ copyEntry  2 }
```

```
sourceType OBJECT-TYPE
SYNTAX INTEGER   {
      host(1),master(2),ep(3)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Source selection for file copy"
::={ copyEntry  3 }



sourceName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Source: host name, IP or MAC address (depending on type of
copy)"
::={ copyEntry  4 }



sourceFileName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Source file name for copying"
::={ copyEntry  5 }



sourceFileType OBJECT-TYPE
SYNTAX INTEGER   {
```

```
      firmware(1),startupconfig(2),runningconfig(3),
backupconfig(4),image(5),boot(6),license(7),dbrunningconfig(8),kernel(9)
,basefs(10)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Source file type for copying (firmware, config, etc)"
::={ copyEntry  6 }



destProtocol OBJECT-TYPE
SYNTAX INTEGER   {
      tftp(3),local(5),http(6),https(7)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Destination file copy protocol (ftp, tftp, scp) - usually
leave blank"
::={ copyEntry  7 }



destType OBJECT-TYPE
SYNTAX INTEGER   {
      host(1),master(2),ep(3),epall(4),masterall(5)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Destination selection for file copy"
::={ copyEntry  8 }



destName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
```

```
STATUS current
DESCRIPTION
"         Destination: host name, IP or MAC address (depending on type
of copy)"
::={ copyEntry  9 }



destFileName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Destination file name for copying"
::={ copyEntry  10 }



destFileType OBJECT-TYPE
SYNTAX INTEGER   {
      firmware(1),startupconfig(2),runningconfig(3),
backupconfig(4),image(5),boot(6),license(7),dbrunningconfig(8),kernel(9)
,basefs(10)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Destination file type for copying (firmware, config, etc)"
::={ copyEntry  11 }



bytesRead OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Bytes read from source file during copy"
::={ copyEntry  12 }
```

```
bytesWrite OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Bytes written to destination file during copy"
::={ copyEntry  13 }



operationStatus OBJECT-TYPE
SYNTAX INTEGER   {
      downloading(1),
applying(2),pending(3),creating(4),uploading(5),non(6)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Status of copy (reading or burning)"
::={ copyEntry  14 }



abort OBJECT-TYPE
SYNTAX TruthValue   {
      true(1),false(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Abort current copy operation"
::={ copyEntry  15 }



start OBJECT-TYPE
SYNTAX TruthValue   {
```

```
      true(1),false(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Start copy transaction"
::={ copyEntry  16 }



copyTableRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         The status of this copy operation"
::={ copyEntry  17 }



generalParamsGroup OBJECT IDENTIFIER ::={ systemGroup  2 }



systemResetToDefault OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reset all system values to factory defaults"
::={ generalParamsGroup  1 }
```

```
systemReboot OBJECT-TYPE
SYNTAX RebootEnum   {
      reboot(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reboot the iMMS"
::={ generalParamsGroup  2 }



hpnaSystemResetToDefault OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reset all HPNA system values to factory defaults"
::={ generalParamsGroup  3 }



sysResetToDefaultKeepConn OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reset all system values to factory defaults except iMMS
connectivity information"
::={ generalParamsGroup  4 }



sysAutoUpgrade OBJECT-TYPE
SYNTAX StartEnum   {
      start(1),noop(2)
```

```
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Auto upgrade the all system . During Upgrade the iMMS , Master
and EP HPNA will be upgraded."
::={ generalParamsGroup  5 }



autoUpgradeSettingsGroup OBJECT IDENTIFIER ::={ generalParamsGroup  6 }



autoUpgradeTimeInterval OBJECT-TYPE
SYNTAX Integer32 ( 300..10080 )
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Time interval in min between testing the new upgrade is
available."
::={ autoUpgradeSettingsGroup  1 }



autoUpgradeTransactionInterval OBJECT-TYPE
SYNTAX Integer32 ( 3..10 )
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Time (in mins) between upgrade of two units"
::={ autoUpgradeSettingsGroup  2 }



autoUpgradeWebPage OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
```

```
DESCRIPTION
"         URL of WEB page where upgrade information is stored"
::={ autoUpgradeSettingsGroup  3 }



autoUpgradeEnable OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         If Auto upgrade feature is enabled"
::={ autoUpgradeSettingsGroup  4 }



topologyGroup OBJECT IDENTIFIER ::={ systemGroup  3 }



ifMasterTopologyTable OBJECT-TYPE
SYNTAX SEQUENCE OF IfMasterTopologyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         HPNA master topology (row per master, per NIC/VLAN)"
::={ topologyGroup  1 }



ifMasterTopologyEntry OBJECT-TYPE
SYNTAX  IfMasterTopologyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Per-master table row"
```

```
INDEX {

          ifMasterTopologyInterfaceId ,
          ifMasterTopologyMasterRFMacAddress
 }
::={ ifMasterTopologyTable  1 }

IfMasterTopologyEntry::= SEQUENCE {

          ifMasterTopologyInterfaceId    Integer32 ,
          ifMasterTopologyMasterRFMacAddress    MacAddress ,
          ifMasterTopologyStatus    INTEGER
 }




ifMasterTopologyInterfaceId OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         HPNA Master interface (NIC/VLAN) id"
::={ ifMasterTopologyEntry  1 }




ifMasterTopologyMasterRFMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         HPNA Master MAC address"
::={ ifMasterTopologyEntry  2 }




ifMasterTopologyStatus OBJECT-TYPE
SYNTAX INTEGER   {
      primary(1),secondary(2)
  }
```

```
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         If master is discovered on several interfaces, one is primary
and other(s) are secondary. All communication is over primary if it is
not down"
::={ ifMasterTopologyEntry  3 }



masterEPTopologyTable OBJECT-TYPE
SYNTAX SEQUENCE OF MasterEPTopologyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Per-Endpoint information table"
::={ topologyGroup  2 }




masterEPTopologyEntry OBJECT-TYPE
SYNTAX  MasterEPTopologyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in the per-endpoint table"
INDEX {

          masterEPTopolgyEPRFMacAddress
 }
::={ masterEPTopologyTable  1 }

MasterEPTopologyEntry::= SEQUENCE {

          masterEPTopolgyMasterRFMacAddress    MacAddress ,
          masterEPTopolgyEPRFMacAddress    MacAddress
 }
```

```
masterEPTopolgyMasterRFMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Master MAC address (on HPNA)"
::={ masterEPTopologyEntry  1 }



masterEPTopolgyEPRFMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Endpoint MAC address (on HPNA)"
::={ masterEPTopologyEntry  2 }



epCPETopologyTable OBJECT-TYPE
SYNTAX SEQUENCE OF EpCPETopologyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Per-Customer information table"
::={ topologyGroup  3 }



epCPETopologyEntry OBJECT-TYPE
SYNTAX  EpCPETopologyEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Per customer information entry"
INDEX {
```

```
          epCPETopologyMacAddress
 }
::={ epCPETopologyTable  1 }

EpCPETopologyEntry::= SEQUENCE {

          epCPETopologyMacAddress    MacAddress ,
          epCPETopologyIPAddress    IpAddress ,
          epCPETopologyEPMacAddress    MacAddress ,
          epCPETopologyOperStatus    INTEGER ,
          epCPETopologyOnlineOnlyInfo    INTEGER
 }



epCPETopologyMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Customer MAC as discovered by Inango system"
::={ epCPETopologyEntry  1 }



epCPETopologyIPAddress OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Customer IP as discovered by Inango system from dhcp messages
or configured by administrator"
::={ epCPETopologyEntry  2 }



epCPETopologyEPMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-create
STATUS current
```

```
DESCRIPTION
"         EP Mac address"
::={ epCPETopologyEntry  3 }



epCPETopologyOperStatus OBJECT-TYPE
SYNTAX INTEGER   {
      offline(1),online(2)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Customer Operational Status"
::={ epCPETopologyEntry  4 }



epCPETopologyOnlineOnlyInfo OBJECT-TYPE
SYNTAX INTEGER   {
      onlineonly(1),showoffline(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Set to ONLINEONLY to delete this endpoints information when it
is offline"
::={ epCPETopologyEntry  6 }



copyHistoryTable OBJECT-TYPE
SYNTAX SEQUENCE OF CopyHistoryEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         History of updates"
::={ systemGroup  4 }
```

```
copyHistoryEntry OBJECT-TYPE
SYNTAX  CopyHistoryEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row per update"
INDEX {

          copyHistoryId
 }
::={ copyHistoryTable  1 }

CopyHistoryEntry::= SEQUENCE {

          copyHistoryId    Integer32 ,
          copyHistoryTimeStamp    DisplayString ,
          copyHistoryStatus    INTEGER ,
          copyHistoryDesc    DisplayString
 }



copyHistoryId OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Id of operation"
::={ copyHistoryEntry  1 }



copyHistoryTimeStamp OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Time update was performed"
```

```
::={ copyHistoryEntry  2 }



copyHistoryStatus OBJECT-TYPE
SYNTAX INTEGER   {
      success(1), failure(2)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Status of update"
::={ copyHistoryEntry  3 }



copyHistoryDesc OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Description"
::={ copyHistoryEntry  4 }



epCustomerTopologyTable OBJECT-TYPE
SYNTAX SEQUENCE OF EpCustomerTopologyEntry
MAX-ACCESS not-accessible
STATUS deprecated
DESCRIPTION
"         Per-Customer information table"
::={ systemGroup  5 }



epCustomerTopologyEntry OBJECT-TYPE
SYNTAX  EpCustomerTopologyEntry
```

```
MAX-ACCESS not-accessible
STATUS deprecated
DESCRIPTION
"         Per customer information entry"
INDEX {

          epCustomerTopologyMacAddress ,
          epCustomerTopologyIPAddress
 }
::={ epCustomerTopologyTable  1 }

EpCustomerTopologyEntry::= SEQUENCE {

          epCustomerTopologyMacAddress    MacAddress ,
          epCustomerTopologyIPAddress    IpAddress ,
          epCustomerTopologyEPMacAddress    MacAddress ,
          epCustomerTopologyOperStatus    INTEGER ,
          epCustomerTopologyOnlineOnlyInfo    INTEGER
 }



epCustomerTopologyMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-create
STATUS deprecated
DESCRIPTION
"         Customer MAC as discovered by the Inango system"
::={ epCustomerTopologyEntry  1 }



epCustomerTopologyIPAddress OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS deprecated
DESCRIPTION
"         Customer IP  as discovered by Inango from dhcp messages or
configured by administrator"
::={ epCustomerTopologyEntry  2 }
```

```
epCustomerTopologyEPMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-create
STATUS deprecated
DESCRIPTION
"         EP Mac address"
::={ epCustomerTopologyEntry  3 }



epCustomerTopologyOperStatus OBJECT-TYPE
SYNTAX INTEGER   {
      offline(1),online(2)
  }
MAX-ACCESS  read-only
STATUS deprecated
DESCRIPTION
"         Customer Operational Status"
::={ epCustomerTopologyEntry  4 }



epCustomerTopologyOnlineOnlyInfo OBJECT-TYPE
SYNTAX INTEGER   {
      onlineonly(1),showoffline(2)
  }
MAX-ACCESS  read-create
STATUS deprecated
DESCRIPTION
"         Should this row deleted when customer is offline"
::={ epCustomerTopologyEntry  6 }



mngGroup OBJECT IDENTIFIER ::={ hpna31MDUdeviceMib  2 }
```

```
interfaceTable OBJECT-TYPE
SYNTAX SEQUENCE OF InterfaceEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Network interfaces on management station"
::={ mngGroup  1 }




interfaceEntry OBJECT-TYPE
SYNTAX  InterfaceEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Per-network interface information on mng station"
INDEX {

          interfaceID
 }
::={ interfaceTable  1 }

InterfaceEntry::= SEQUENCE {

          interfaceID    Integer32 ,
          interfaceType    INTEGER ,
          interfaceAlias    DisplayString ,
          interfaceName    DisplayString ,
          adminStatus    INTEGER ,
          operStatus    INTEGER ,
          ifaceMngEnable    EnableEnum ,
          ifaceHPNAScaningEnable    EnableEnum ,
          ifaceMacAddress    MacAddress
 }



interfaceID OBJECT-TYPE
```

```
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Automatically generated during interface id"
::={ interfaceEntry  1 }



interfaceType OBJECT-TYPE
SYNTAX INTEGER   {
      vlan(1),nic(2)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Interface type during VLAN or NIC"
::={ interfaceEntry  2 }



interfaceAlias OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         User-assigned interface name (optional)"
::={ interfaceEntry  3 }



interfaceName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Automatically generated interface name"
::={ interfaceEntry  4 }
```

```
adminStatus OBJECT-TYPE
SYNTAX INTEGER   {
      down(1), up(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         User-specified interface status (up or down)"
::={ interfaceEntry  5 }



operStatus OBJECT-TYPE
SYNTAX INTEGER   {
      down(1), up(2)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Current interface status (up or down)"
::={ interfaceEntry  6 }



ifaceMngEnable OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Set to ENABLE in order to management over this iMMS interface"
::={ interfaceEntry  8 }



ifaceHPNAScaningEnable OBJECT-TYPE
SYNTAX EnableEnum   {
```

```
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Discover masters (topology) by scanning this iMMS interface"
::={ interfaceEntry  9 }



ifaceMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         iMMS Interface Mac Address"
::={ interfaceEntry  10 }



ipAddressTable OBJECT-TYPE
SYNTAX SEQUENCE OF IpAddressEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Table for storing IP information for each interface"
::={ mngGroup  2 }



ipAddressEntry OBJECT-TYPE
SYNTAX  IpAddressEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Per IP address information"
INDEX {

          ifID
```

```
 }
::={ ipAddressTable  1 }

IpAddressEntry::= SEQUENCE {

          ipAddress    IpAddress ,
          mask    IpAddress ,
          priority    Integer32 ,
          type    INTEGER ,
          ifID    Integer32 ,
          ipAddressRowStatus    RowStatus
 }



ipAddress OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         IP address (IPv4)"
::={ ipAddressEntry  1 }



mask OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         IP address mask"
::={ ipAddressEntry  2 }



priority OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
```

```
"         If multiple IP address entries exist on a system, this defines
the priority (metric) of this entry. Pri 0 is selected if active, etc."
::={ ipAddressEntry  4 }



type OBJECT-TYPE
SYNTAX INTEGER   {
      static(1),dhcp(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         IP address may be statically or DHCP-assigned"
::={ ipAddressEntry  5 }



ifID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Interface id (matches an entry in interfaceTable)"
::={ ipAddressEntry  6 }



ipAddressRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         The status of this conceptual row."
::={ ipAddressEntry  7 }
```

```
iMMSVlanGroup OBJECT IDENTIFIER ::={ mngGroup  3 }



vlanTable OBJECT-TYPE
SYNTAX SEQUENCE OF VlanEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         iMMS VLAN-specific information"
::={ iMMSVlanGroup  1 }




vlanEntry OBJECT-TYPE
SYNTAX  VlanEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in vlanTable"
INDEX {

          vlanVID
 }
::={ vlanTable  1 }

VlanEntry::= SEQUENCE {

          vlanVID    Integer32 ,
          vlanAlias    DisplayString ,
          vlanRowStatus    RowStatus ,
          vlanInterfaceID    Integer32
 }



vlanVID OBJECT-TYPE
```

```
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         VLAN id for this iMMS VLAN"
::={ vlanEntry  1 }



vlanAlias OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         User-assigned vlan name (optional)"
::={ vlanEntry  2 }



vlanRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Status of this VLAN"
::={ vlanEntry  3 }



vlanInterfaceID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         ID of VLAN interface"
::={ vlanEntry  4 }
```

```
vlanPortTable OBJECT-TYPE
SYNTAX SEQUENCE OF VlanPortEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Per-VLAN information"
::={ iMMSVlanGroup  2 }




vlanPortEntry OBJECT-TYPE
SYNTAX  VlanPortEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in Per-VLAN table"
INDEX {

          vlanPortVID ,
          portInterfaceID
 }
::={ vlanPortTable  1 }

VlanPortEntry::= SEQUENCE {

          vlanPortVID    Integer32 ,
          portInterfaceID    Integer32 ,
          vlanPortTagged    INTEGER ,
          vlanPortRowStatus    RowStatus
 }




vlanPortVID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
```

```
DESCRIPTION
"         VLAN tag (id) value. Range is 1-4095."
::={ vlanPortEntry  1 }



portInterfaceID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Port Interface ID belonging to this VLAN from Interface Table"
::={ vlanPortEntry  2 }



vlanPortTagged OBJECT-TYPE
SYNTAX INTEGER   {
      tagged(1),untagged(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Defines whether traffic on this NIC  is tagged or untagged"
::={ vlanPortEntry  3 }



vlanPortRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         The status of this conceptual row."
::={ vlanPortEntry  4 }
```

```
userManagmentTable OBJECT-TYPE
SYNTAX SEQUENCE OF UserManagmentEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Users that are authorized to log in to management system"
::={ mngGroup  4 }



userManagmentEntry OBJECT-TYPE
SYNTAX  UserManagmentEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Per-user information"
INDEX {

          userName
 }
::={ userManagmentTable  1 }

UserManagmentEntry::= SEQUENCE {

          userName    DisplayString ,
          password    DisplayString ,
          userAccessPermissions    INTEGER ,
          userManagmentTableRowStatus    RowStatus
 }



userName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         User name"
```

```
::={ userManagmentEntry  1 }



password OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         User password"
::={ userManagmentEntry  2 }



userAccessPermissions OBJECT-TYPE
SYNTAX INTEGER   {
      review(1),config(2),admin(3)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         User access permissions (review, config, admin)"
::={ userManagmentEntry  3 }



userManagmentTableRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         The status of this row"
::={ userManagmentEntry  4 }
```

```
iMMSDiagnosticGroup OBJECT IDENTIFIER ::={ mngGroup  5 }



iMMSDiagnosticControlTable OBJECT-TYPE
SYNTAX SEQUENCE OF IMMSDiagnosticControlEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Control (create, configure and run) diagnostics tests on an
interface"
::={ iMMSDiagnosticGroup  1 }




iMMSDiagnosticControlEntry OBJECT-TYPE
SYNTAX  IMMSDiagnosticControlEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in table - corresponds to a specific diagnostic test"
INDEX {

          iMMSDiagnosticControlID
 }
::={ iMMSDiagnosticControlTable  1 }

IMMSDiagnosticControlEntry::= SEQUENCE {

          iMMSDiagnosticControlID    Integer32 ,
          iMMSDiagnosticControlOperation    INTEGER ,
          iMMSDiagnosticControlHostName    DisplayString ,
          iMMSDiagnosticControlLastReceivedTime    DisplayString ,
          iMMSDiagnosticControlResult    INTEGER ,
          iMMSDiagnosticControlRowStatus    RowStatus ,
          iMMSDiagnosticControlOperationArguments    DisplayString
 }
```

```
iMMSDiagnosticControlID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         ID for this diagostic test"
::={ iMMSDiagnosticControlEntry  1 }



iMMSDiagnosticControlOperation OBJECT-TYPE
SYNTAX INTEGER   {
      ping(1),traceroute(2),arp(3)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Type of test to run (ping, traceroute or arp)"
::={ iMMSDiagnosticControlEntry  2 }



iMMSDiagnosticControlHostName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Name (or IP address) of unit to which ping/traceroute/arp is
sent"
::={ iMMSDiagnosticControlEntry  3 }



iMMSDiagnosticControlLastReceivedTime OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Time of last test"
```

```
::={ iMMSDiagnosticControlEntry  4 }



iMMSDiagnosticControlResult OBJECT-TYPE
SYNTAX INTEGER   {
      indiag(1),noop(2)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Result of last diagnostic test"
::={ iMMSDiagnosticControlEntry  5 }



iMMSDiagnosticControlRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Status of row"
::={ iMMSDiagnosticControlEntry  6 }



iMMSDiagnosticControlOperationArguments OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Arguments passed to diagnostic operation"
::={ iMMSDiagnosticControlEntry  7 }
```

```
iMMSDiagnosticInfoTable OBJECT-TYPE
SYNTAX SEQUENCE OF IMMSDiagnosticInfoEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         This table contains the results of diagnostics tests"
::={ iMMSDiagnosticGroup  2 }




iMMSDiagnosticInfoEntry OBJECT-TYPE
SYNTAX  IMMSDiagnosticInfoEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry (row) in diagnostics results table"
INDEX {

          iMMSDiagnosticInfoSessionID ,
          iMMSDiagnosticInfoInfoID
 }
::={ iMMSDiagnosticInfoTable  1 }

IMMSDiagnosticInfoEntry::= SEQUENCE {

          iMMSDiagnosticInfoSessionID    Integer32 ,
          iMMSDiagnosticInfoInfoID    Integer32 ,
          iMMSDiagnosticInfoResult    DisplayString
 }




iMMSDiagnosticInfoSessionID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Diagnostic test identifier"
::={ iMMSDiagnosticInfoEntry  1 }
```

```
iMMSDiagnosticInfoInfoID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         For looped tests (e.g. ping) - this is the loop number"
::={ iMMSDiagnosticInfoEntry  2 }



iMMSDiagnosticInfoResult OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Result from designated loop of this test"
::={ iMMSDiagnosticInfoEntry  3 }



iMMSSettings OBJECT IDENTIFIER ::={ mngGroup  6 }



snmpSettingsGroup OBJECT IDENTIFIER ::={ iMMSSettings  1 }



snmpPublicCommunityString OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         SNMP Public Community"
::={ snmpSettingsGroup  1 }
```

```
snmpPrivateCommunityString OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         SNMP Private Community"
::={ snmpSettingsGroup  2 }



snmpTrapServersTable OBJECT-TYPE
SYNTAX SEQUENCE OF SnmpTrapServersEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Table of TRAP Servers"
::={ snmpSettingsGroup  3 }




snmpTrapServersEntry OBJECT-TYPE
SYNTAX  SnmpTrapServersEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in Trap Server table"
INDEX {

          snmpTrapServersID
 }
::={ snmpTrapServersTable  1 }

SnmpTrapServersEntry::= SEQUENCE {

          snmpTrapServersID    Integer32 ,
          snmpTrapServersHostName    DisplayString ,
          snmpTrapServersPriority    Integer32 ,
```

```
          snmpTrapServersRowStatus    RowStatus
 }



snmpTrapServersID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Trap Server ID"
::={ snmpTrapServersEntry  1 }



snmpTrapServersHostName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Host Name of Trap Server"
::={ snmpTrapServersEntry  2 }



snmpTrapServersPriority OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Trap Server Priority"
::={ snmpTrapServersEntry  3 }



snmpTrapServersRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
```

```
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Row status"
::={ snmpTrapServersEntry  4 }



snmpSysLocation OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         SNMP Location"
::={ snmpSettingsGroup  4 }



snmpSysContact OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         SNMP Contact info"
::={ snmpSettingsGroup  5 }



webSettingsGroup OBJECT IDENTIFIER ::={ iMMSSettings  2 }



webLinePerTable OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Maximum number of lines displayed on each screen per table"
```

```
::={ webSettingsGroup  1 }



webInfoDirection OBJECT-TYPE
SYNTAX INTEGER   {
      vertical(1),horizontal(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         When true the direction of information is vertical, when false
the direction is horizontal"
::={ webSettingsGroup  2 }



logSettingsGroup OBJECT IDENTIFIER ::={ iMMSSettings  4 }



localLogTable OBJECT-TYPE
SYNTAX SEQUENCE OF LocalLogEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Table of logs"
::={ logSettingsGroup  1 }



localLogEntry OBJECT-TYPE
SYNTAX  LocalLogEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in system log tyable - i.e. an individual event log"
INDEX {
```

```
          localLogMessageID
 }
::={ localLogTable  1 }

LocalLogEntry::= SEQUENCE {

          localLogMessageID    Integer32 ,
          localLogTimestamp    DisplayString ,
          localLogModuleID    Integer32 ,
          localLogMsgLevel    INTEGER ,
          localLogMsgMnemonic    DisplayString ,
          localLogMsg    DisplayString
 }



localLogMessageID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Identified of event log"
::={ localLogEntry  1 }



localLogTimestamp OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Time of event log"
::={ localLogEntry  2 }



localLogModuleID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
```

```
STATUS current
DESCRIPTION
"         Module that generated the event log"
::={ localLogEntry  3 }



localLogMsgLevel OBJECT-TYPE
SYNTAX INTEGER   {

emergency(0),alert(1),critical(2),error(3),warning(4),notice(5),informat
ional(6),debug(7)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Urgency of event log: emergency,  alert, critical, error,
warning, notice, informational, or debug"
::={ localLogEntry  4 }



localLogMsgMnemonic OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Log message mnemonic"
::={ localLogEntry  5 }



localLogMsg OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Log message mnemonic"
::={ localLogEntry  6 }
```

```
localLogMsgMaxLevel OBJECT-TYPE
SYNTAX INTEGER   {

emergency(0),alert(1),critical(2),error(3),warning(4),notice(5),informat
ional(6),debug(7)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Only store system logs for events up to this level of urgency"
::={ logSettingsGroup  2 }



localLogMsgMaxNumber OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Maximum number of system logs to store"
::={ logSettingsGroup  3 }



localLogMsgIgnores OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         The number of log messages which were ignored.  A message will
be ignored if it has a severity value greater than defined maximum"
::={ logSettingsGroup  4 }



localLogMsgEnable OBJECT-TYPE
SYNTAX EnableEnum   {
```

```
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Set to ENABLE to turn on message logging"
::={ logSettingsGroup  5 }



localLogMsgClean OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Clear system event log"
::={ logSettingsGroup  6 }



syslogServerConfigTable OBJECT-TYPE
SYNTAX SEQUENCE OF SyslogServerConfigEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Syslog server information"
::={ logSettingsGroup  7 }



syslogServerConfigEntry OBJECT-TYPE
SYNTAX  SyslogServerConfigEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in syslog server table"
INDEX {
```

```
          syslogServerID
 }
::={ syslogServerConfigTable  1 }

SyslogServerConfigEntry::= SEQUENCE {

          syslogServerID    Integer32 ,
          syslogServerIPAddress    IpAddress ,
          syslogServerEnable    EnableEnum ,
          syslogServerRowStatus    RowStatus
 }



syslogServerID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Syslog server identified"
::={ syslogServerConfigEntry  1 }



syslogServerIPAddress OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         IP address of syslog server"
::={ syslogServerConfigEntry  2 }



syslogServerEnable OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-create
```

```
STATUS current
DESCRIPTION
"         Set to ENABLE to enable logging to this syslog server"
::={ syslogServerConfigEntry  3 }



syslogServerRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Row status"
::={ syslogServerConfigEntry  4 }



syslogMsgMaxLevel OBJECT-TYPE
SYNTAX INTEGER   {

emergency(0),alert(1),critical(2),error(3),warning(4),notice(5),informat
ional(6),debug(7)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Log events up to this level of urgency"
::={ logSettingsGroup  8 }



syslogMsgIgnores OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
```

```
"         Number of messages ignored (urgency number above defined
maximum)"
::={ logSettingsGroup  9 }



iMMSHostname OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Host name of iMMS system"
::={ mngGroup  9 }



timeGroup OBJECT IDENTIFIER ::={ mngGroup  10 }



timeZone OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         This parameter define the local time zone"
::={ timeGroup  1 }



timeFormat OBJECT-TYPE
SYNTAX INTEGER   {
      yyyymmddhms(1), hms(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Format in which to display time (extended or brief)"
::={ timeGroup  2 }
```

```
localTime OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Local time"
::={ timeGroup  3 }



ntpClientEnable OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Set to ENABLE to activate this NTP client"
::={ timeGroup  4 }



ntpServerTable OBJECT-TYPE
SYNTAX SEQUENCE OF NtpServerEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         NTP Server information table"
::={ timeGroup  5 }



ntpServerEntry OBJECT-TYPE
SYNTAX  NtpServerEntry
MAX-ACCESS not-accessible
```

```
STATUS current
DESCRIPTION
"         Entry (row) in NTP Server table"
INDEX {

          ntpServerId
 }
::={ ntpServerTable  1 }

NtpServerEntry::= SEQUENCE {

          ntpServerId    Integer32 ,
          ntpServerHostName    DisplayString ,
          ntpServerPriority    Integer32 ,
          ntpServerRowStatus    RowStatus
 }



ntpServerId OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         NTP Server ID"
::={ ntpServerEntry  1 }



ntpServerHostName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Host name for machine on which NTP Server runs"
::={ ntpServerEntry  2 }



ntpServerPriority OBJECT-TYPE
```

```
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Priority of NTP Server"
::={ ntpServerEntry  3 }



ntpServerRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Row status"
::={ ntpServerEntry  4 }



iMMSLicenseInfo OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         IMMS license data"
::={ mngGroup  11 }



iMMSSWRelease OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         iMMS SW release name"
::={ mngGroup  12 }
```

```
defaultGateway OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         iMMS default gateway"
::={ mngGroup  16 }



primaryDNSServer OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Primary DNS Server"
::={ mngGroup  14 }



secondaryDNSServer OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Secondary DNS Server"
::={ mngGroup  15 }



iMMSDhcpRelayGroup OBJECT IDENTIFIER ::={ mngGroup  17 }



iMMSDhcpRelayEnable OBJECT-TYPE
SYNTAX EnableEnum   {
```

```
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Set to ENABLE to enable DHCP Relay on iMMS"
::={ iMMSDhcpRelayGroup  1 }



iMMSDhcpRelaySrvListTable OBJECT-TYPE
SYNTAX SEQUENCE OF IMMSDhcpRelaySrvListEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Remote DHCP Server table"
::={ iMMSDhcpRelayGroup  2 }




iMMSDhcpRelaySrvListEntry OBJECT-TYPE
SYNTAX  IMMSDhcpRelaySrvListEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in remote DHCP Server Table"
INDEX {

          iMMSDhcpRelaySrvListServerID
 }
::={ iMMSDhcpRelaySrvListTable  1 }

IMMSDhcpRelaySrvListEntry::= SEQUENCE {

          iMMSDhcpRelaySrvListServerID    Integer32 ,
          iMMSDhcpRelaySrvListIPAddress    IpAddress ,
          iMMSDhcpRelaySrvListPriority    Integer32 ,
          iMMSDhcpRelaySrvListRowStatus    RowStatus
 }
```

```
iMMSDhcpRelaySrvListServerID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         ID of remote DHCP Server"
::={ iMMSDhcpRelaySrvListEntry  1 }



iMMSDhcpRelaySrvListIPAddress OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         IP Address (or hostname) of remote DHCP Server"
::={ iMMSDhcpRelaySrvListEntry  2 }



iMMSDhcpRelaySrvListPriority OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Priority of remote DHCP Server (use highest number first)"
::={ iMMSDhcpRelaySrvListEntry  3 }



iMMSDhcpRelaySrvListRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
```

```
DESCRIPTION
"         Row status"
::={ iMMSDhcpRelaySrvListEntry  4 }



iMMSDhcpRelayMasterTable OBJECT-TYPE
SYNTAX SEQUENCE OF IMMSDhcpRelayMasterEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Configuration parameters for DHCP Relay operation"
::={ iMMSDhcpRelayGroup  3 }




iMMSDhcpRelayMasterEntry OBJECT-TYPE
SYNTAX  IMMSDhcpRelayMasterEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in DHCP Relay configuration table"
INDEX {

          iMMSDhcpRelayMasterMacAddr
 }
::={ iMMSDhcpRelayMasterTable  1 }

IMMSDhcpRelayMasterEntry::= SEQUENCE {

          iMMSDhcpRelayMasterMacAddr    MacAddress ,
          iMMSDhcpRelayMasterEnable    EnableEnum ,
          iMMSDhcpRelayMasterRemoteIDType    INTEGER ,
          iMMSDhcpRelayMasterRemoteIDString    DisplayString ,
          iMMSDhcpRelayMasterCircuitIdType    INTEGER ,
          iMMSDhcpRelayMasterCircuitIdString    DisplayString
 }
```

```
iMMSDhcpRelayMasterMacAddr OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         HPNA MAC address of HPNA Master unit for which DHCP Relay is
being done"
::={ iMMSDhcpRelayMasterEntry  1 }



iMMSDhcpRelayMasterEnable OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Enable DHCP Relay for this HPNA Master"
::={ iMMSDhcpRelayMasterEntry  2 }



iMMSDhcpRelayMasterRemoteIDType OBJECT-TYPE
SYNTAX INTEGER   {
      epMac(1),masterMac(2),userDefined(3)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Value to be used in DHCP Relay RemoteID field (endpoint MAC,
master MAC or user-defined)"
::={ iMMSDhcpRelayMasterEntry  3 }



iMMSDhcpRelayMasterRemoteIDString OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
```

```
STATUS current
DESCRIPTION
"         If RemoteID is user-define, use this value in DHCP Relay
messages (20 chars)"
::={ iMMSDhcpRelayMasterEntry  4 }



iMMSDhcpRelayMasterCircuitIdType OBJECT-TYPE
SYNTAX INTEGER   {
      epMac(1),masterMac(2),userDefined(3)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Value to be used in DHCP Relay CircuitID field (endpoint MAC,
master MAC or user-defined)"
::={ iMMSDhcpRelayMasterEntry  5 }



iMMSDhcpRelayMasterCircuitIdString OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         If CircuitID is user-define, use this value in DHCP Relay
messages (20 chars)"
::={ iMMSDhcpRelayMasterEntry  6 }



iMMSHWInfo OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Information about CPU where the iMMS is running"
::={ mngGroup  18 }
```

```
iMMSRoutingGroup OBJECT IDENTIFIER ::={ mngGroup  19 }



iMMSRoutingTable OBJECT-TYPE
SYNTAX SEQUENCE OF IMMSRoutingEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         iMMS Routing Table"
::={ iMMSRoutingGroup  1 }




iMMSRoutingEntry OBJECT-TYPE
SYNTAX  IMMSRoutingEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in iMMS Routing table"
INDEX {

          iMMSRoutingEntryID
 }
::={ iMMSRoutingTable  1 }

IMMSRoutingEntry::= SEQUENCE {

          iMMSRoutingEntryID    Integer32 ,
          iMMSRoutingDest    IpAddress ,
          iMMSRoutingMask    IpAddress ,
          iMMSRoutingNH    IpAddress ,
          iMMSRoutingIf    Integer32 ,
          iMMSRoutingMetric    Integer32 ,
          iMMSRoutingType    INTEGER ,
          iMMSRoutingRowStatus    RowStatus
 }
```

```
iMMSRoutingEntryID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Routing table entry ID"
::={ iMMSRoutingEntry  1 }



iMMSRoutingDest OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Destination IP address"
::={ iMMSRoutingEntry  2 }



iMMSRoutingMask OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Routing mask"
::={ iMMSRoutingEntry  3 }



iMMSRoutingNH OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Next hop"
```

```
::={ iMMSRoutingEntry  4 }



iMMSRoutingIf OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Interface ID"
::={ iMMSRoutingEntry  5 }



iMMSRoutingMetric OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Priority"
::={ iMMSRoutingEntry  6 }



iMMSRoutingType OBJECT-TYPE
SYNTAX INTEGER   {
      static(1),dynamic(2)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Static (user-defined) or dynamic (learned)"
::={ iMMSRoutingEntry  7 }



iMMSRoutingRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
```

```
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Row status"
::={ iMMSRoutingEntry  14 }



iMMSArpTable OBJECT-TYPE
SYNTAX SEQUENCE OF IMMSArpEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         ARP  table"
::={ iMMSRoutingGroup  2 }



iMMSArpEntry OBJECT-TYPE
SYNTAX  IMMSArpEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in ARP Proxy table"
INDEX {

          iMMSArpEntryID
 }
::={ iMMSArpTable  1 }

IMMSArpEntry::= SEQUENCE {

          iMMSArpEntryID    Integer32 ,
          iMMSArpIpAddr    IpAddress ,
          iMMSArpMacAddr    MacAddress ,
          iMMSArpIf    Integer32 ,
          iMMSArpType    INTEGER ,
```

```
          iMMSArpRowStatus    RowStatus
 }



iMMSArpEntryID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Identifier for entry in ARP Proxy table"
::={ iMMSArpEntry  1 }



iMMSArpIpAddr OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         IP address"
::={ iMMSArpEntry  2 }



iMMSArpMacAddr OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         MAC address"
::={ iMMSArpEntry  3 }



iMMSArpIf OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
```

```
DESCRIPTION
"         Interface"
::={ iMMSArpEntry  4 }



iMMSArpType OBJECT-TYPE
SYNTAX INTEGER   {
      static(1),dynamic(2)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         ARP type (dynamic/static)"
::={ iMMSArpEntry  5 }



iMMSArpRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Row status"
::={ iMMSArpEntry  8 }



authenticationGroup OBJECT IDENTIFIER ::={ mngGroup  20 }



authenticatorGroup OBJECT IDENTIFIER ::={ authenticationGroup  1 }
```

```
eapReauthPeriod OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reauthantication period ( min)"
::={ authenticatorGroup  1 }



eapolVersion OBJECT-TYPE
SYNTAX INTEGER   {
      ver1(1),ver2(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         IEEE 802.1X/EAPOL version. The eapol is implemented based on
IEEE Std 802.1X-2004 which defines EAPOL version 2. However, there are
many client implementations that do not handle the new version number
correctly (they seem to drop the frames completely). In order to make
hostapd interoperate with these clients, the version number can be set
to the older version (1)."
::={ authenticatorGroup  2 }



eapolIpAddr OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         The own IP address of the master (used as NAS-IP-Address). By
default ip address configured on interface 1."
::={ authenticatorGroup  3 }



nasIdentifier OBJECT-TYPE
```

```
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Optional NAS-Identifier string for RADIUS messages. When used,
this should be a unique to the NAS within the scope of the RADIUS
server. For example, a fully qualified domain name can be used here.By
default Master Mac address."
::={ authenticatorGroup  4 }



primaryAuthServerAddr OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         RADIUS authentication server"
::={ authenticatorGroup  5 }



primaryAuthServerPort OBJECT-TYPE
SYNTAX Integer32 ( 1..65535 )
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Authentication port . Default 1812."
::={ authenticatorGroup  6 }



primaryAuthServerSharedSecret OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Shared Secret for Primary  Radius Server"
::={ authenticatorGroup  7 }
```

```
secondaryAuthServerAddr OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         RADIUS authentication server"
::={ authenticatorGroup  8 }



secondaryAuthServerPort OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Authentication port . Default 1812."
::={ authenticatorGroup  9 }



secondaryAuthServerSharedSecret OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Shared Secret for Secondary Radius Server"
::={ authenticatorGroup  10 }



acctServerAddr OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         RADIUS accounting server"
::={ authenticatorGroup  11 }
```

```
acctServerPort OBJECT-TYPE
SYNTAX Integer32 ( 1..65535 )
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Default 1813."
::={ authenticatorGroup  12 }



acctServerSharedSecret OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Shared Secret for Accounting Server"
::={ authenticatorGroup  13 }



supplicantGroup OBJECT IDENTIFIER ::={ authenticationGroup  2 }



enableAuthenticator OBJECT-TYPE
SYNTAX TruthValue   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Enable Authenticator"
::={ authenticationGroup  3 }
```

```
enableSupplicant OBJECT-TYPE
SYNTAX TruthValue   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Enable Supplicant"
::={ authenticationGroup  4 }



masterGroup OBJECT IDENTIFIER ::={ hpna31MDUdeviceMib  3 }



masterInfoTable OBJECT-TYPE
SYNTAX SEQUENCE OF MasterInfoEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         HPNA Master information table"
::={ masterGroup  1 }




masterInfoEntry OBJECT-TYPE
SYNTAX  MasterInfoEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in per-master table"
INDEX {

          masterRFMacAddress
 }
::={ masterInfoTable  1 }
```

```
MasterInfoEntry::= SEQUENCE {

          masterRFMacAddress    MacAddress ,
          manufacturerInfo    DisplayString ,
          modelNumber    DisplayString ,
          serialNumber    DisplayString ,
          softwareVersion    DisplayString ,
          vendorSpecialInfo    DisplayString ,
          reboot    RebootEnum ,
          masterInfoStatus    INTEGER ,
          aliasName    DisplayString ,
          resetToDefault    ResetEnum ,
          resetMasterSubSystemToDefault    ResetEnum ,
          masterInfoRowStatus    RowStatus ,
          masterInfoAuth    INTEGER ,
          masterLAGEnable    EnableEnum ,
          masterInfoFWSig    DisplayString
 }



masterRFMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Master MAC address (on HPNA)"
::={ masterInfoEntry  1 }



manufacturerInfo OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Manufacturer name"
::={ masterInfoEntry  2 }
```

```
modelNumber OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Product model number"
::={ masterInfoEntry  3 }



serialNumber OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Product serial number"
::={ masterInfoEntry  4 }



softwareVersion OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         HPNA Master firmware version"
::={ masterInfoEntry  5 }



vendorSpecialInfo OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Vendor-specific information"
::={ masterInfoEntry  6 }
```

```
reboot OBJECT-TYPE
SYNTAX RebootEnum   {
      reboot(1),noop(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Reboot the master"
::={ masterInfoEntry  7 }



masterInfoStatus OBJECT-TYPE
SYNTAX INTEGER   {
      online(1),offline(2),error(3)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Master status (offline/online from management system)"
::={ masterInfoEntry  8 }



aliasName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         User-specific name for this master"
::={ masterInfoEntry  9 }



resetToDefault OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
```

```
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Reset all values on master to factory defaults"
::={ masterInfoEntry  10 }



resetMasterSubSystemToDefault OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Reset master and all connected endpoints to factory defaults"
::={ masterInfoEntry  11 }



masterInfoRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         The status of this conceptual row."
::={ masterInfoEntry  12 }



masterInfoAuth OBJECT-TYPE
SYNTAX INTEGER   {
      authorized(1),nonauthorized(2)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
```

```
"         If this master is authorized as Inango Master"
::={ masterInfoEntry  13 }



masterLAGEnable OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Enable Link Aggregation (LAG)"
::={ masterInfoEntry  14 }



masterInfoFWSig OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Signature of FW image for master hpna chip"
::={ masterInfoEntry  15 }



masterDiagnosticsGroup OBJECT IDENTIFIER ::={ masterGroup  2 }



masterDiagnosticsRFTable OBJECT-TYPE
SYNTAX SEQUENCE OF MasterDiagnosticsRFEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         ChangeMe"
::={ masterDiagnosticsGroup  1 }
```

```
masterDiagnosticsRFEntry OBJECT-TYPE
SYNTAX  MasterDiagnosticsRFEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         ChangeMe"
INDEX {

          masterDiagnosticsRFMacAddress
 }
::={ masterDiagnosticsRFTable  1 }

MasterDiagnosticsRFEntry::= SEQUENCE {

          masterDiagnosticsRFMacAddress    MacAddress ,
          masterDiagnosticsRFStartTest    TruthValue ,
          masterDiagnosticsRFPacketNumberInTest    Integer32 ,
          masterDiagnosticsRFResetResults    ResetEnum ,
          masterDiagnosticsRFResetLigthResults    ResetEnum
 }



masterDiagnosticsRFMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Master MAC address (on HPNA)"
::={ masterDiagnosticsRFEntry  1 }



masterDiagnosticsRFStartTest OBJECT-TYPE
SYNTAX TruthValue   {
      true(1),false(2)
  }
```

```
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Diagnostics test may be performed now (or not)"
::={ masterDiagnosticsRFEntry  2 }



masterDiagnosticsRFPacketNumberInTest OBJECT-TYPE
SYNTAX Integer32 ( 100..1200 )
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Number of packets to be sent in diagnostics test"
::={ masterDiagnosticsRFEntry  3 }



masterDiagnosticsRFResetResults OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reset/clean results from previous run"
::={ masterDiagnosticsRFEntry  4 }



masterDiagnosticsRFResetLigthResults OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reset/clean results from previous run for ligth diagnostics"
::={ masterDiagnosticsRFEntry  5 }
```

```
masterDiagnosticsXTalkTable OBJECT-TYPE
SYNTAX SEQUENCE OF MasterDiagnosticsXTalkEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Master cross talk diagnostics table"
::={ masterDiagnosticsGroup  2 }




masterDiagnosticsXTalkEntry OBJECT-TYPE
SYNTAX  MasterDiagnosticsXTalkEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in Master Cross Talk Diagnostics table"
INDEX {

          masterXTalkTestMacAddress
 }
::={ masterDiagnosticsXTalkTable  1 }

MasterDiagnosticsXTalkEntry::= SEQUENCE {

          masterXTalkTestMacAddress    MacAddress ,
          masterXTalkTestLastTestPerformed    DisplayString ,
          masterXTalkTestStatus    INTEGER ,
          masterXTalkTestStartTest    TruthValue ,
          masterXTalkTestClearResults    ResetEnum ,
          masterXTalkTestCrossPackets    Integer32 ,
          masterXTalkTestCrossMac1    MacAddress ,
          masterXTalkTestCrossMac2    MacAddress ,
          masterXTalkTestCrossMac3    MacAddress ,
          masterXTalkTestCrossMac4    MacAddress ,
          masterXTalkTestCrossMac5    MacAddress
 }
```

```
masterXTalkTestMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Master HPNA MAC Address"
::={ masterDiagnosticsXTalkEntry  1 }



masterXTalkTestLastTestPerformed OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Time when test was performed the last time on the master"
::={ masterDiagnosticsXTalkEntry  2 }



masterXTalkTestStatus OBJECT-TYPE
SYNTAX INTEGER   {
      intest(1),non(2),error(3)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Status of diagnostic test (intest, non,error)"
::={ masterDiagnosticsXTalkEntry  3 }



masterXTalkTestStartTest OBJECT-TYPE
SYNTAX TruthValue   {
      true(1),false(2)
  }
MAX-ACCESS  read-write
STATUS current
```

```
DESCRIPTION
"         Start now test on the master"
::={ masterDiagnosticsXTalkEntry  4 }



masterXTalkTestClearResults OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Clear all  xtalk information gathered on this device"
::={ masterDiagnosticsXTalkEntry  5 }



masterXTalkTestCrossPackets OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Number of packets received by the master from another master
(cross packets)"
::={ masterDiagnosticsXTalkEntry  6 }



masterXTalkTestCrossMac1 OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Source MAC addr 1 of cross packets"
::={ masterDiagnosticsXTalkEntry  8 }
```

```
masterXTalkTestCrossMac2 OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Source MAC addr 2 of cross packets"
::={ masterDiagnosticsXTalkEntry  9 }



masterXTalkTestCrossMac3 OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Source MAC addr 3 of cross packets"
::={ masterDiagnosticsXTalkEntry  10 }



masterXTalkTestCrossMac4 OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Source MAC addr 4 of cross packets"
::={ masterDiagnosticsXTalkEntry  11 }



masterXTalkTestCrossMac5 OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Source MAC addr 5 of cross packets"
::={ masterDiagnosticsXTalkEntry  12 }
```

```
masterNoiseTestHPNATable OBJECT-TYPE
SYNTAX SEQUENCE OF MasterNoiseTestHPNAEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Master HPNA noise test table"
::={ masterDiagnosticsGroup  3 }




masterNoiseTestHPNAEntry OBJECT-TYPE
SYNTAX  MasterNoiseTestHPNAEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in Master HPNA noise test table"
INDEX {

          masterNoiseTestHPNAMacAddress
 }
::={ masterNoiseTestHPNATable  1 }

MasterNoiseTestHPNAEntry::= SEQUENCE {

          masterNoiseTestHPNAMacAddress    MacAddress ,
          masterNoiseTestHPNALastTestPerformed    DisplayString ,
          masterNoiseTestHPNAStatus    Integer32 ,
          masterNoiseTestHPNAStartTest    TruthValue ,
          masterNoiseTestHPNACleanResults    ResetEnum ,
          masterNoiseTestHPNANumOfIterations    INTEGER
 }




masterNoiseTestHPNAMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
```

```
"         Master HPNA Mac Address"
::={ masterNoiseTestHPNAEntry  1 }



masterNoiseTestHPNALastTestPerformed OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Time the last master noise test was performed"
::={ masterNoiseTestHPNAEntry  2 }



masterNoiseTestHPNAStatus OBJECT-TYPE
SYNTAX Integer32   {
      intest(1), non(2), error(3)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Status of noise test (intest, non,error)"
::={ masterNoiseTestHPNAEntry  3 }



masterNoiseTestHPNAStartTest OBJECT-TYPE
SYNTAX TruthValue   {
      true(1),false(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Start noise test on the Master."
::={ masterNoiseTestHPNAEntry  4 }
```

```
masterNoiseTestHPNACleanResults OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Clean noise test results of the master"
::={ masterNoiseTestHPNAEntry  5 }



masterNoiseTestHPNANumOfIterations OBJECT-TYPE
SYNTAX INTEGER ( 2..8 )
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Number of iterations of getting measurements from the device.
Default value is 4."
::={ masterNoiseTestHPNAEntry  6 }



masterStatisticGroup OBJECT IDENTIFIER ::={ masterGroup  3 }



masterStatisticRFTable OBJECT-TYPE
SYNTAX SEQUENCE OF MasterStatisticRFEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Master HPNA statistics table"
::={ masterStatisticGroup  1 }



masterStatisticRFEntry OBJECT-TYPE
```

```
SYNTAX  MasterStatisticRFEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in Masters HPNA statistics table"
INDEX {

          masterStatisticRFMacAddress
 }
::={ masterStatisticRFTable  1 }

MasterStatisticRFEntry::= SEQUENCE {

          masterStatisticRFMacAddress    MacAddress ,
          masterStatisticRF1HTxUcastPackets    Counter64 ,
          masterStatisticRF1HTxMcastPackets    Counter64 ,
          masterStatisticRF1HTxBcastPackets    Counter64 ,
          masterStatisticRF1HTxBytes    Counter64 ,
          masterStatisticRF1HRxUcastPackets    Counter64 ,
          masterStatisticRF1HRxMcastPackets    Counter64 ,
          masterStatisticRF1HRxBcastPackets    Counter64 ,
          masterStatisticRF1HRxBytes    Counter64 ,
          masterStatisticRF12HTxUcastPackets    Counter64 ,
          masterStatisticRF12HTxMcastPackets    Counter64 ,
          masterStatisticRF12HTxBcastPackets    Counter64 ,
          masterStatisticRF12HTxBytes    Counter64 ,
          masterStatisticRF12HRxUcastPackets    Counter64 ,
          masterStatisticRF12HRxMcastPackets    Counter64 ,
          masterStatisticRF12HRxBcastPackets    Counter64 ,
          masterStatisticRF12HRxBytes    Counter64 ,
          masterStatisticRF24HTxUcastPackets    Counter64 ,
          masterStatisticRF24HTxMcastPackets    Counter64 ,
          masterStatisticRF24HTxBcastPackets    Counter64 ,
          masterStatisticRF24HTxBytes    Counter64 ,
          masterStatisticRF24HRxUcastPackets    Counter64 ,
          masterStatisticRF24HRxMcastPackets    Counter64 ,
          masterStatisticRF24HRxBcastPackets    Counter64 ,
          masterStatisticRF24HRxBytes    Counter64 ,
          masterStatisticRFLastReadTxUcastPackets    Counter64 ,
          masterStatisticRFLastReadTxMcastPackets    Counter64 ,
          masterStatisticRFLastReadTxBcastPackets    Counter64 ,
          masterStatisticRFLastReadTxBytes    Counter64 ,
          masterStatisticRFLastReadRxUcastPackets    Counter64 ,
```

```
        masterStatisticRFLastReadRxMcastPackets    Counter64 ,
        masterStatisticRFLastReadRxBcastPackets    Counter64 ,
        masterStatisticRFLastReadRxBytes    Counter64 ,
        masterStatisticRFAggregatedTxUcastPackets    Counter64 ,
        masterStatisticRFAggregatedTxMcastPackets    Counter64 ,
        masterStatisticRFAggregatedTxBcastPackets    Counter64 ,
        masterStatisticRFAggregatedTxBytes    Counter64 ,
        masterStatisticRFAggregatedRxUcastPackets    Counter64 ,
        masterStatisticRFAggregatedRxMcastPackets    Counter64 ,
        masterStatisticRFAggregatedRxBcastPackets    Counter64 ,
        masterStatisticRFAggregatedRxBytes    Counter64 ,
        masterStatisticRFResetCounters    ResetEnum
 }



masterStatisticRFMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Master MAC address (on HPNA)"
::={ masterStatisticRFEntry  1 }



masterStatisticRF1HTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during last 1 hour"
::={ masterStatisticRFEntry  2 }



masterStatisticRF1HTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
```

```
DESCRIPTION
"         Multicast packets sent during last 1 hour"
::={ masterStatisticRFEntry  3 }



masterStatisticRF1HTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during last 1 hour"
::={ masterStatisticRFEntry  4 }



masterStatisticRF1HTxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes sent during last 1 hour"
::={ masterStatisticRFEntry  5 }



masterStatisticRF1HRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets received during last 1 hour"
::={ masterStatisticRFEntry  8 }



masterStatisticRF1HRxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
```

```
STATUS current
DESCRIPTION
"         Multicast packets received during last 1 hour"
::={ masterStatisticRFEntry  9 }



masterStatisticRF1HRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during last 1 hour"
::={ masterStatisticRFEntry  10 }



masterStatisticRF1HRxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during last 1 hour"
::={ masterStatisticRFEntry  11 }



masterStatisticRF12HTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during last 12 hour"
::={ masterStatisticRFEntry  14 }



masterStatisticRF12HTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
```

```
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets sent during last 12 hour"
::={ masterStatisticRFEntry  15 }



masterStatisticRF12HTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during last 12 hour"
::={ masterStatisticRFEntry  16 }



masterStatisticRF12HTxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes sent during last 12 hour"
::={ masterStatisticRFEntry  17 }



masterStatisticRF12HRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets received during last 12 hour"
::={ masterStatisticRFEntry  20 }



masterStatisticRF12HRxMcastPackets OBJECT-TYPE
```

```
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets received during last 12 hour"
::={ masterStatisticRFEntry  21 }



masterStatisticRF12HRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during last 12 hour"
::={ masterStatisticRFEntry  22 }



masterStatisticRF12HRxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during last 12 hour"
::={ masterStatisticRFEntry  23 }



masterStatisticRF24HTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during last 24 hour"
::={ masterStatisticRFEntry  26 }
```

```
masterStatisticRF24HTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets sent during last 24 hour"
::={ masterStatisticRFEntry  27 }



masterStatisticRF24HTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during last 24 hour"
::={ masterStatisticRFEntry  28 }



masterStatisticRF24HTxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes sent during last 24 hour"
::={ masterStatisticRFEntry  29 }



masterStatisticRF24HRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets received during last 24 hour"
::={ masterStatisticRFEntry  32 }
```

```
masterStatisticRF24HRxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets received during last 24 hour"
::={ masterStatisticRFEntry  33 }



masterStatisticRF24HRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during last 24 hour"
::={ masterStatisticRFEntry  34 }



masterStatisticRF24HRxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during last 24 hour"
::={ masterStatisticRFEntry  35 }



masterStatisticRFLastReadTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during last number polled"
::={ masterStatisticRFEntry  38 }
```

```
masterStatisticRFLastReadTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets sent during last number polled"
::={ masterStatisticRFEntry  39 }
```

```
masterStatisticRFLastReadTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during last number polled"
::={ masterStatisticRFEntry  40 }
```

```
masterStatisticRFLastReadTxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes sent during last number polled"
::={ masterStatisticRFEntry  41 }
```

```
masterStatisticRFLastReadRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during last number polled"
::={ masterStatisticRFEntry  44 }
```

```
masterStatisticRFLastReadRxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets received during last number polled"
::={ masterStatisticRFEntry  45 }
```

```
masterStatisticRFLastReadRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during last number polled"
::={ masterStatisticRFEntry  46 }
```

```
masterStatisticRFLastReadRxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during last number polled"
::={ masterStatisticRFEntry  47 }
```

```
masterStatisticRFAggregatedTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during aggregate since deployment"
::={ masterStatisticRFEntry  50 }
```

```
masterStatisticRFAggregatedTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets sent during aggregate since deployment"
::={ masterStatisticRFEntry  51 }



masterStatisticRFAggregatedTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during aggregate since deployment"
::={ masterStatisticRFEntry  52 }



masterStatisticRFAggregatedTxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes sent during aggregate since deployment"
::={ masterStatisticRFEntry  53 }



masterStatisticRFAggregatedRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets received during aggregate since deployment"
```

```
::={ masterStatisticRFEntry  56 }



masterStatisticRFAggregatedRxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets received during aggregate since deployment"
::={ masterStatisticRFEntry  57 }



masterStatisticRFAggregatedRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during aggregate since deployment"
::={ masterStatisticRFEntry  58 }



masterStatisticRFAggregatedRxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during aggregate since deployment"
::={ masterStatisticRFEntry  59 }



masterStatisticRFResetCounters OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
```

```
STATUS current
DESCRIPTION
"         Reset counters on HPNA interface"
::={ masterStatisticRFEntry  63 }



resetAllMasterStatistics OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reset statistics counters of all masters of the system"
::={ masterStatisticGroup  3 }



masterEPDevicesTable OBJECT-TYPE
SYNTAX SEQUENCE OF MasterEPDevicesEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         TBD"
::={ masterStatisticGroup  4 }



masterEPDevicesEntry OBJECT-TYPE
SYNTAX  MasterEPDevicesEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         TBD"
INDEX {

          masterEPDeviceMacAddress
 }
```

```
::={ masterEPDevicesTable  1 }

MasterEPDevicesEntry::= SEQUENCE {

          masterEPDeviceMacAddress    MacAddress ,
          masterEPDeviceEPSeen    Integer32 ,
          masterEPDeviceEPOnline    Integer32 ,
          masterEPDeviceEPAuthorized    Integer32 ,
          masterEPDeviceEPNonAuthorized    Integer32 ,
          masterEPDeviceHostsOnline    Integer32
 }



masterEPDeviceMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Master for which ep devices and host information is presented"
::={ masterEPDevicesEntry  1 }



masterEPDeviceEPSeen OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Number of EPs which were seen on this master after last system
reset"
::={ masterEPDevicesEntry  2 }



masterEPDeviceEPOnline OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
```

```
"         Number of EPs online"
::={ masterEPDevicesEntry  3 }



masterEPDeviceEPAuthorized OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Number of Authorized EPs"
::={ masterEPDevicesEntry  4 }



masterEPDeviceEPNonAuthorized OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Number of Non-Authorized EPs"
::={ masterEPDevicesEntry  5 }



masterEPDeviceHostsOnline OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Number of Hosts Online"
::={ masterEPDevicesEntry  6 }



masterMacMcastGroup OBJECT IDENTIFIER ::={ masterGroup  4 }
```

```
masterMcastTable OBJECT-TYPE
SYNTAX SEQUENCE OF MasterMcastEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Multi-cast table"
::={ masterMacMcastGroup  1 }




masterMcastEntry OBJECT-TYPE
SYNTAX  MasterMcastEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in Master Multicast table"
INDEX {

          masterMcastMacAddress
 }
::={ masterMcastTable  1 }

MasterMcastEntry::= SEQUENCE {

          masterMcastMacAddress    MacAddress ,
          masterMcastIGMPSnoopingEnable    EnableEnum ,
          masterMcastIGMPSnoopingTimeout    Integer32
 }




masterMcastMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Master for which max mcast groups is defined"
::={ masterMcastEntry  1 }
```

```
masterMcastIGMPSnoopingEnable OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Enable IGMP Snooping"
::={ masterMcastEntry  3 }



masterMcastIGMPSnoopingTimeout OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         IGMP Snooping timeout (seconds)"
::={ masterMcastEntry  4 }



epGroup OBJECT IDENTIFIER ::={ hpna31MDUdeviceMib  4 }



epInfoTable OBJECT-TYPE
SYNTAX SEQUENCE OF EpInfoEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Endpoint information table"
::={ epGroup  1 }
```

```
epInfoEntry OBJECT-TYPE
SYNTAX  EpInfoEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Per-endpoint entry"
INDEX {

          epInfoRFMacAddress
 }
::={ epInfoTable  1 }

EpInfoEntry::= SEQUENCE {

          epInfoRFMacAddress    MacAddress ,
          epInfoManufactoryInfo    DisplayString ,
          epInfoModelNumber    DisplayString ,
          epInfoSerialNumber    DisplayString ,
          epInfoSoftwareVersion    DisplayString ,
          epInfoVendorSpecialInfo    DisplayString ,
          epInfoReboot    RebootEnum ,
          epInfoInfoStatus    INTEGER ,
          epInfoAliasName    DisplayString ,
          epInfoResetToDefault    ResetEnum ,
          epInfoNumberOfNetPort    Integer32 ,
          epInfoTrafficEnabled    TruthValue ,
          epInfoMaxNumberOfHosts    Integer32 ,
          epInfoIgmpSnoopingEnabled    TruthValue ,
          epInfoRowStatus    RowStatus ,
          epInfoAuth    INTEGER ,
          epInfoFWSig    DisplayString
 }



epInfoRFMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Endpoint HPNA MAC address"
```

```
::={ epInfoEntry  1 }



epInfoManufactoryInfo OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Endpoint manufacturer information"
::={ epInfoEntry  2 }



epInfoModelNumber OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Endpoint model number"
::={ epInfoEntry  3 }



epInfoSerialNumber OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Endpoint serial number"
::={ epInfoEntry  4 }



epInfoSoftwareVersion OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
```

```
"         Endpoint firmware version"
::={ epInfoEntry  5 }



epInfoVendorSpecialInfo OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Endpoint vendor-specific information"
::={ epInfoEntry  6 }



epInfoReboot OBJECT-TYPE
SYNTAX RebootEnum   {
      reboot(1),noop(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Reboot endpoint"
::={ epInfoEntry  7 }



epInfoInfoStatus OBJECT-TYPE
SYNTAX INTEGER   {
      online(1),offline(2),error(3)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Endpoint status (online, offline)"
::={ epInfoEntry  8 }
```

```
epInfoAliasName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         User-specified name for this endpoint"
::={ epInfoEntry  9 }



epInfoResetToDefault OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Reset endpoint parameters to factory defaults"
::={ epInfoEntry  10 }



epInfoNumberOfNetPort OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Number of Ethernet interfaces"
::={ epInfoEntry  11 }



epInfoTrafficEnabled OBJECT-TYPE
SYNTAX TruthValue   {
      true(1),false(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Enable traffic to pass through this endpoint or not (boolean)"
```

```
::={ epInfoEntry  12 }



epInfoMaxNumberOfHosts OBJECT-TYPE
SYNTAX Integer32 ( 1..10 )
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Maximum number of computers (hosts) that can be connected to
this endpoint"
::={ epInfoEntry  13 }



epInfoIgmpSnoopingEnabled OBJECT-TYPE
SYNTAX TruthValue   {
      true(1),false(2)
  }
MAX-ACCESS  read-create
STATUS deprecated
DESCRIPTION
"         Enable IGMP Snooping or not (boolean)"
::={ epInfoEntry  14 }



epInfoRowStatus OBJECT-TYPE
SYNTAX RowStatus   {
      active(1), notInService(2), notReady(3), createAndGo(4),
createAndWait(5), destroy(6)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Row status"
::={ epInfoEntry  15 }
```

```
epInfoAuth OBJECT-TYPE
SYNTAX INTEGER   {
      authorized(1),nonauthorized(2)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Endpoint information authorization"
::={ epInfoEntry  16 }



epInfoFWSig OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Signature of FW image for ep  hpna chip"
::={ epInfoEntry  17 }



epDiagnosticGroup OBJECT IDENTIFIER ::={ epGroup  2 }



epDiagnosticHPNATable OBJECT-TYPE
SYNTAX SEQUENCE OF EpDiagnosticHPNAEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Endpoint HPNA diagnostics table"
::={ epDiagnosticGroup  1 }



epDiagnosticHPNAEntry OBJECT-TYPE
```

```
SYNTAX  EpDiagnosticHPNAEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in endpoint HPNA diagnostics table"
INDEX {

          epDiagnosticHPNAMacAddress
 }
::={ epDiagnosticHPNATable  1 }

EpDiagnosticHPNAEntry::= SEQUENCE {

          epDiagnosticHPNAMacAddress    MacAddress ,
          epDiagnosticHPNALastTestPerformed    DisplayString ,
          epDiagnosticHPNAStatus    INTEGER ,
          epDiagnosticHPNADownstreamPer    DisplayString ,
          epDiagnosticHPNADownstreamSnr    DisplayString ,
          epDiagnosticHPNADownstreamRate    DisplayString ,
          epDiagnosticHPNADownstreamPower    DisplayString ,
          epDiagnosticHPNADownstreamPhyRate    DisplayString ,
          epDiagnosticHPNAUpstreamPer    DisplayString ,
          epDiagnosticHPNAUpstreamSnr    DisplayString ,
          epDiagnosticHPNAUpstreamRate    DisplayString ,
          epDiagnosticHPNAUpstreamPower    DisplayString ,
          epDiagnosticHPNAUpstreamPhyRate    DisplayString
 }



epDiagnosticHPNAMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Endpoint MAC address"
::={ epDiagnosticHPNAEntry  1 }



epDiagnosticHPNALastTestPerformed OBJECT-TYPE
```

```
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Time last diagnostic test was performed"
::={ epDiagnosticHPNAEntry  2 }



epDiagnosticHPNAStatus OBJECT-TYPE
SYNTAX INTEGER   {
      intest(1),non(2),error(3)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Status of diagnostic test (intest, non)"
::={ epDiagnosticHPNAEntry  3 }



epDiagnosticHPNADownstreamPer OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Downstream Packet Error Rate"
::={ epDiagnosticHPNAEntry  6 }



epDiagnosticHPNADownstreamSnr OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Downstream Signal-Noise Ration"
::={ epDiagnosticHPNAEntry  7 }
```

```
epDiagnosticHPNADownstreamRate OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Downstream rate"
::={ epDiagnosticHPNAEntry  8 }



epDiagnosticHPNADownstreamPower OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Downstream power"
::={ epDiagnosticHPNAEntry  9 }



epDiagnosticHPNADownstreamPhyRate OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Downstream PHY rate"
::={ epDiagnosticHPNAEntry  10 }



epDiagnosticHPNAUpstreamPer OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Upstream Packet Error Rate"
::={ epDiagnosticHPNAEntry  11 }
```

```
epDiagnosticHPNAUpstreamSnr OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Upstream Signal-Noise Ration"
::={ epDiagnosticHPNAEntry  12 }
```

```
epDiagnosticHPNAUpstreamRate OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Upstream rate"
::={ epDiagnosticHPNAEntry  13 }
```

```
epDiagnosticHPNAUpstreamPower OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Upstream power"
::={ epDiagnosticHPNAEntry  14 }
```

```
epDiagnosticHPNAUpstreamPhyRate OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Upstream PHY bit rate"
::={ epDiagnosticHPNAEntry  16 }
```

```
epLightDiagnosticHPNATable OBJECT-TYPE
SYNTAX SEQUENCE OF EpLightDiagnosticHPNAEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Table of light HPNA endpoint diagnostics information"
::={ epDiagnosticGroup  3 }




epLightDiagnosticHPNAEntry OBJECT-TYPE
SYNTAX  EpLightDiagnosticHPNAEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in light endpoint HPNA diagnostics table"
INDEX {

          epLightDiagnosticHPNAMacAddress
 }
::={ epLightDiagnosticHPNATable  1 }

EpLightDiagnosticHPNAEntry::= SEQUENCE {

          epLightDiagnosticHPNAMacAddress    MacAddress ,
          epLightDiagnosticHPNALastTestPerformed    DisplayString ,
          epLightDiagnosticHPNAStatus    INTEGER ,
          epLightDiagnosticHPNAStartTest    TruthValue ,
          epLightDiagnosticHPNAPacketNumberInTest    Integer32 ,
          epLightDiagnosticHPNADownstreamSnr    DisplayString ,
          epLightDiagnosticHPNADownstreamRate    DisplayString ,
          epLightDiagnosticHPNADownstreamPower    DisplayString ,
          epLightDiagnosticHPNADownstreamPhyRate    DisplayString ,
          epLightDiagnosticHPNAUpstreamSnr    DisplayString ,
          epLightDiagnosticHPNAUpstreamRate    DisplayString ,
          epLightDiagnosticHPNAUpstreamPower    DisplayString ,
          epLightDiagnosticHPNAUpstreamPhyRate    DisplayString
```

```
  }



epLightDiagnosticHPNAMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Endpoint HPNA MAC address"
::={ epLightDiagnosticHPNAEntry  1 }



epLightDiagnosticHPNALastTestPerformed OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Time the last EP diagnostics test was performed"
::={ epLightDiagnosticHPNAEntry  2 }



epLightDiagnosticHPNAStatus OBJECT-TYPE
SYNTAX INTEGER   {
      intest(1),non(2),error(3)
  }
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Status of diagnostic test (intest, non,error)"
::={ epLightDiagnosticHPNAEntry  3 }



epLightDiagnosticHPNAStartTest OBJECT-TYPE
SYNTAX TruthValue   {
      true(1),false(2)
```

```
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Diagnostics test may be performed now (or not)"
::={ epLightDiagnosticHPNAEntry  4 }



epLightDiagnosticHPNAPacketNumberInTest OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Number of packets to be sent in diagnostics test"
::={ epLightDiagnosticHPNAEntry  5 }



epLightDiagnosticHPNADownstreamSnr OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Downstream Signal-Noise Ratio"
::={ epLightDiagnosticHPNAEntry  6 }



epLightDiagnosticHPNADownstreamRate OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Downstream data rate"
::={ epLightDiagnosticHPNAEntry  7 }
```

```
epLightDiagnosticHPNADownstreamPower OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Downstream power"
::={ epLightDiagnosticHPNAEntry  8 }



epLightDiagnosticHPNADownstreamPhyRate OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Downstream PHY rate"
::={ epLightDiagnosticHPNAEntry  9 }



epLightDiagnosticHPNAUpstreamSnr OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Upstream Signal-Noise Ratio"
::={ epLightDiagnosticHPNAEntry  10 }



epLightDiagnosticHPNAUpstreamRate OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Upstream data rate"
::={ epLightDiagnosticHPNAEntry  11 }
```

```
epLightDiagnosticHPNAUpstreamPower OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Upstream power"
::={ epLightDiagnosticHPNAEntry  12 }



epLightDiagnosticHPNAUpstreamPhyRate OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Upstream PHY bit rate"
::={ epLightDiagnosticHPNAEntry  14 }



epNoiseTestHPNATable OBJECT-TYPE
SYNTAX SEQUENCE OF EpNoiseTestHPNAEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Table of HPNA noise test activating"
::={ epDiagnosticGroup  4 }



epNoiseTestHPNAEntry OBJECT-TYPE
SYNTAX  EpNoiseTestHPNAEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in EP noise test table"
INDEX {
```

```
          epNoiseTestHPNAMacAddress
 }
::={ epNoiseTestHPNATable  1 }

EpNoiseTestHPNAEntry::= SEQUENCE {

          epNoiseTestHPNAMacAddress    MacAddress ,
          epNoiseTestHPNALastTestPerformed    DisplayString ,
          epNoiseTestHPNAStatus    INTEGER ,
          epNoiseTestHPNAStartTest    TruthValue ,
          epNoiseTestHPNACleanResults    ResetEnum ,
          epNoiseTestHPNANumOfIterations    INTEGER
 }



epNoiseTestHPNAMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Endpoint HPNA MAC address"
::={ epNoiseTestHPNAEntry  1 }



epNoiseTestHPNALastTestPerformed OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Time the last EP noise test was performed"
::={ epNoiseTestHPNAEntry  2 }



epNoiseTestHPNAStatus OBJECT-TYPE
SYNTAX INTEGER   {
      intest(1),non(2),error(3)
  }
```

```
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Status of noise test (intest, non,error)"
::={ epNoiseTestHPNAEntry  3 }



epNoiseTestHPNAStartTest OBJECT-TYPE
SYNTAX TruthValue   {
      true(1),false(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Start noise test on EP."
::={ epNoiseTestHPNAEntry  4 }



epNoiseTestHPNACleanResults OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Clean noise test results of the ep"
::={ epNoiseTestHPNAEntry  5 }



epNoiseTestHPNANumOfIterations OBJECT-TYPE
SYNTAX INTEGER
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Number of iterations of getting measurement from the device.
Default value is 4."
::={ epNoiseTestHPNAEntry  6 }
```

```
cleanAllEPDiagnostics OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Clean diagnostics info of all EPs of the system"
::={ epDiagnosticGroup  5 }



cleanAllEPLightDiagnostics OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Clean light diagnostics info of all EPs of the system"
::={ epDiagnosticGroup  6 }



cleanAllEPNoiseTestResults OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Clean noise test info of all EPs of the system"
::={ epDiagnosticGroup  7 }
```

```
epStatisticGroup OBJECT IDENTIFIER ::={ epGroup  3 }



epStatisticRFTable OBJECT-TYPE
SYNTAX SEQUENCE OF EpStatisticRFEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Endpoint statistics table"
::={ epStatisticGroup  1 }




epStatisticRFEntry OBJECT-TYPE
SYNTAX  EpStatisticRFEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in endpoint statistics table"
INDEX {

          epStatisticRFMacAddress
 }
::={ epStatisticRFTable  1 }

EpStatisticRFEntry::= SEQUENCE {

          epStatisticRFMacAddress    MacAddress ,
          epStatisticRF1HTxUcastPackets    Counter64 ,
          epStatisticRF1HTxMcastPackets    Counter64 ,
          epStatisticRF1HTxBcastPackets    Counter64 ,
          epStatisticRF1HTxBytes    Counter64 ,
          epStatisticRF1HRxUcastPackets    Counter64 ,
          epStatisticRF1HRxMcastPackets    Counter64 ,
          epStatisticRF1HRxBcastPackets    Counter64 ,
          epStatisticRF1HRxBytes    Counter64 ,
          epStatisticRF12HTxUcastPackets    Counter64 ,
          epStatisticRF12HTxMcastPackets    Counter64 ,
          epStatisticRF12HTxBcastPackets    Counter64 ,
```

```
            epStatisticRF12HTxBytes    Counter64 ,
            epStatisticRF12HRxUcastPackets    Counter64 ,
            epStatisticRF12HRxMcastPackets    Counter64 ,
            epStatisticRF12HRxBcastPackets    Counter64 ,
            epStatisticRF12HRxBytes    Counter64 ,
            epStatisticRF24HTxUcastPackets    Counter64 ,
            epStatisticRF24HTxMcastPackets    Counter64 ,
            epStatisticRF24HTxBcastPackets    Counter64 ,
            epStatisticRF24HTxBytes    Counter64 ,
            epStatisticRF24HRxUcastPackets    Counter64 ,
            epStatisticRF24HRxMcastPackets    Counter64 ,
            epStatisticRF24HRxBcastPackets    Counter64 ,
            epStatisticRF24HRxBytes    Counter64 ,
            epStatisticRFLastReadTxUcastPackets    Counter64 ,
            epStatisticRFLastReadTxMcastPackets    Counter64 ,
            epStatisticRFLastReadTxBcastPackets    Counter64 ,
            epStatisticRFLastReadTxBytes    Counter64 ,
            epStatisticRFLastReadRxUcastPackets    Counter64 ,
            epStatisticRFLastReadRxMcastPackets    Counter64 ,
            epStatisticRFLastReadRxBcastPackets    Counter64 ,
            epStatisticRFLastReadRxBytes    Counter64 ,
            epStatisticRFAggregatedTxUcastPackets    Counter64 ,
            epStatisticRFAggregatedTxMcastPackets    Counter64 ,
            epStatisticRFAggregatedTxBcastPackets    Counter64 ,
            epStatisticRFAggregatedTxBytes    Counter64 ,
            epStatisticRFAggregatedRxUcastPackets    Counter64 ,
            epStatisticRFAggregatedRxMcastPackets    Counter64 ,
            epStatisticRFAggregatedRxBcastPackets    Counter64 ,
            epStatisticRFAggregatedRxBytes    Counter64 ,
            epStatisticRFResetCounters    ResetEnum
 }



epStatisticRFMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         HPNA-MAC address for this endpoint"
::={ epStatisticRFEntry  1 }
```

```
epStatisticRF1HTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during last 1 hour"
::={ epStatisticRFEntry  2 }
```

```
epStatisticRF1HTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets sent during last 1 hour"
::={ epStatisticRFEntry  3 }
```

```
epStatisticRF1HTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during last 1 hour"
::={ epStatisticRFEntry  4 }
```

```
epStatisticRF1HTxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes sent during last 1 hour"
::={ epStatisticRFEntry  5 }
```

```
epStatisticRF1HRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets received during last 1 hour"
::={ epStatisticRFEntry  8 }



epStatisticRF1HRxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets received during last 1 hour"
::={ epStatisticRFEntry  9 }



epStatisticRF1HRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during last 1 hour"
::={ epStatisticRFEntry  10 }



epStatisticRF1HRxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during last 1 hour"
```

```
::={ epStatisticRFEntry  11 }



epStatisticRF12HTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during last 12 hour"
::={ epStatisticRFEntry  14 }



epStatisticRF12HTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets sent during last 12 hour"
::={ epStatisticRFEntry  15 }



epStatisticRF12HTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during last 12 hour"
::={ epStatisticRFEntry  16 }



epStatisticRF12HTxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
```

```
"         Unicast bytes sent during last 12 hour"
::={ epStatisticRFEntry  17 }



epStatisticRF12HRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets received during last 12 hour"
::={ epStatisticRFEntry  20 }



epStatisticRF12HRxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets received during last 12 hour"
::={ epStatisticRFEntry  21 }



epStatisticRF12HRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during last 12 hour"
::={ epStatisticRFEntry  22 }



epStatisticRF12HRxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
```

```
DESCRIPTION
"         Unicast bytes received during last 12 hour"
::={ epStatisticRFEntry  23 }



epStatisticRF24HTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during last 24 hour"
::={ epStatisticRFEntry  26 }



epStatisticRF24HTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets sent during last 24 hour"
::={ epStatisticRFEntry  27 }



epStatisticRF24HTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during last 24 hour"
::={ epStatisticRFEntry  28 }



epStatisticRF24HTxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
```

```
STATUS current
DESCRIPTION
"         Unicast bytes sent during last 24 hour"
::={ epStatisticRFEntry  29 }



epStatisticRF24HRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets received during last 24 hour"
::={ epStatisticRFEntry  32 }



epStatisticRF24HRxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets received during last 24 hour"
::={ epStatisticRFEntry  33 }



epStatisticRF24HRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during last 24 hour"
::={ epStatisticRFEntry  34 }



epStatisticRF24HRxBytes OBJECT-TYPE
SYNTAX Counter64
```

```
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during last 24 hour"
::={ epStatisticRFEntry  35 }



epStatisticRFLastReadTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during last number polled"
::={ epStatisticRFEntry  38 }



epStatisticRFLastReadTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets sent during last number polled"
::={ epStatisticRFEntry  39 }



epStatisticRFLastReadTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during last number polled"
::={ epStatisticRFEntry  40 }



epStatisticRFLastReadTxBytes OBJECT-TYPE
```

```
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes sent during last number polled"
::={ epStatisticRFEntry  41 }



epStatisticRFLastReadRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during last number polled"
::={ epStatisticRFEntry  44 }



epStatisticRFLastReadRxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets received during last number polled"
::={ epStatisticRFEntry  45 }



epStatisticRFLastReadRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during last number polled"
::={ epStatisticRFEntry  46 }
```

```
epStatisticRFLastReadRxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during last number polled"
::={ epStatisticRFEntry  47 }



epStatisticRFAggregatedTxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets sent during aggregate since deployment"
::={ epStatisticRFEntry  50 }



epStatisticRFAggregatedTxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets sent during aggregate since deployment"
::={ epStatisticRFEntry  51 }



epStatisticRFAggregatedTxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets sent during aggregate since deployment"
::={ epStatisticRFEntry  52 }
```

```
epStatisticRFAggregatedTxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes sent during aggregate since deployment"
::={ epStatisticRFEntry  53 }



epStatisticRFAggregatedRxUcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast packets received during aggregate since deployment"
::={ epStatisticRFEntry  56 }



epStatisticRFAggregatedRxMcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Multicast packets received during aggregate since deployment"
::={ epStatisticRFEntry  57 }



epStatisticRFAggregatedRxBcastPackets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Broadcast packets received during aggregate since deployment"
::={ epStatisticRFEntry  58 }
```

```
epStatisticRFAggregatedRxBytes OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS  read-only
STATUS current
DESCRIPTION
"         Unicast bytes received during aggregate since deployment"
::={ epStatisticRFEntry  59 }



epStatisticRFResetCounters OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reset counters on HPNA interface on endpoint"
::={ epStatisticRFEntry  63 }



resetAllEPStatistics OBJECT-TYPE
SYNTAX ResetEnum   {
      reset(1),noop(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Reset statistics counters of all EPs of the system"
::={ epStatisticGroup  3 }



epMcastGroup OBJECT IDENTIFIER ::={ epGroup  4 }
```

```
epMcastTable OBJECT-TYPE
SYNTAX SEQUENCE OF EpMcastEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Max number of mcast flow table"
::={ epMcastGroup  1 }




epMcastEntry OBJECT-TYPE
SYNTAX  EpMcastEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in multicast  endpoint table"
INDEX {

          epMcastMacAddress
 }
::={ epMcastTable  1 }

EpMcastEntry::= SEQUENCE {

          epMcastMacAddress    MacAddress ,
          epMcastIGMPSnoopingEnable    EnableEnum ,
          epMcastIGMPSnoopingTimeout    Integer32
 }




epMcastMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         MAC address of endpoint in max mcast/ep table"
::={ epMcastEntry  1 }
```

```
epMcastIGMPSnoopingEnable OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         Enable IGMP Snooping on endpoint"
::={ epMcastEntry  3 }



epMcastIGMPSnoopingTimeout OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
"         IGMP Snooping timeout (in seconds)"
::={ epMcastEntry  4 }



epRateLimitationGroup OBJECT IDENTIFIER ::={ epGroup  5 }



epHPNARateLimitationTable OBJECT-TYPE
SYNTAX SEQUENCE OF EpHPNARateLimitationEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Endpoint rate limitation table"
::={ epRateLimitationGroup  1 }
```

```
epHPNARateLimitationEntry OBJECT-TYPE
SYNTAX  EpHPNARateLimitationEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in endpoint rate limitation table"
INDEX {

          epHPNARateLimitationMacAddress
 }
::={ epHPNARateLimitationTable  1 }

EpHPNARateLimitationEntry::= SEQUENCE {

          epHPNARateLimitationMacAddress    MacAddress ,
          epHPNAUCUpstreamLimit    Integer32 ,
          epHPNABCUpstreamLimit    Integer32 ,
          epHPNAMCUpstreamLimit    Integer32 ,
          epHPNATotalDownstreamLimit    Integer32 ,
          epHPNATotalDownstreamPeakLimit    Integer32
 }



epHPNARateLimitationMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         MAC address of endpoint"
::={ epHPNARateLimitationEntry  1 }



epHPNAUCUpstreamLimit OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Upstream Unicast rate limit (Kbps)"
```

```
::={ epHPNARateLimitationEntry  5 }



epHPNABCUpstreamLimit OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Upstream broadcast rate limit (Kbps)"
::={ epHPNARateLimitationEntry  6 }



epHPNAMCUpstreamLimit OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Upstream Multicast rate limit (Kbps)"
::={ epHPNARateLimitationEntry  7 }



epHPNATotalDownstreamLimit OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Total upstream limit"
::={ epHPNARateLimitationEntry  14 }



epHPNATotalDownstreamPeakLimit OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
```

```
"         Total downstream peak rate"
::={ epHPNARateLimitationEntry  16 }



epMacGroup OBJECT IDENTIFIER ::={ epGroup  7 }



epHPNAMacTable OBJECT-TYPE
SYNTAX SEQUENCE OF EpHPNAMacEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         MAC address table"
::={ epMacGroup  1 }




epHPNAMacEntry OBJECT-TYPE
SYNTAX  EpHPNAMacEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Row in MAC address table"
INDEX {

          epHPNAMacMacAddress
 }
::={ epHPNAMacTable  1 }

EpHPNAMacEntry::= SEQUENCE {

          epHPNAMacMacAddress    MacAddress ,
          epHPNAMacVlanEnable    EnableEnum ,
          epHPNAMacVlanTag    Integer32 ,
          epHPNAMacVlanOtherTagPassDrop    EnableEnum
 }
```

```
epHPNAMacMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         MAC address for this endpoint"
::={ epHPNAMacEntry  1 }



epHPNAMacVlanEnable OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Enable/disable VLANs on this endpoint"
::={ epHPNAMacEntry  2 }



epHPNAMacVlanTag OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Vlan ID for this endpoint over RF"
::={ epHPNAMacEntry  3 }



epHPNAMacVlanOtherTagPassDrop OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-create
```

```
STATUS current
DESCRIPTION
"         Undefined VLANs pass through this endpoint unchanged (boolean.
If FALSE frame is dropped)"
::={ epHPNAMacEntry  6 }



epNetMacTable OBJECT-TYPE
SYNTAX SEQUENCE OF EpNetMacEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Interface table on Ethernet ports"
::={ epMacGroup  2 }




epNetMacEntry OBJECT-TYPE
SYNTAX  EpNetMacEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"         Entry in Interface table on Ethernet port"
INDEX {

          epNetMacMacAddress ,
          epNetMacPortID
 }
::={ epNetMacTable  1 }

EpNetMacEntry::= SEQUENCE {

          epNetMacMacAddress    MacAddress ,
          epNetMacPortID    Integer32 ,
          epNetMacVlanTag    Integer32 ,
          epNetMacPortTaged    INTEGER ,
          epNetMacVlanTagRemove    EnableEnum
 }
```

```
epNetMacMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         MAC address"
::={ epNetMacEntry  1 }



epNetMacPortID OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS   not-accessible
STATUS current
DESCRIPTION
"         Port number"
::={ epNetMacEntry  2 }



epNetMacVlanTag OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Vlan ID set on network port"
::={ epNetMacEntry  3 }



epNetMacPortTaged OBJECT-TYPE
SYNTAX INTEGER   {
      tagged(1),untagged(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
```

```
"         Port is tagged in VLAN or not"
::={ epNetMacEntry  4 }



epNetMacVlanTagRemove OBJECT-TYPE
SYNTAX EnableEnum   {
      enable(1),disable(2)
  }
MAX-ACCESS  read-create
STATUS current
DESCRIPTION
"         Remove tag on outgoing traffic or leave"
::={ epNetMacEntry  5 }


--------------------
--Start TRAP section
--------------------
epTrapGroup OBJECT IDENTIFIER ::={hpna31MDUnotification 1 }

epDiscoveredTrap NOTIFICATION-TYPE
OBJECTS { masterEPTopolgyMasterRFMacAddress,
masterEPTopolgyEPRFMacAddress,
epInfoAuth}
STATUS current
DESCRIPTION
" New EP is discovered "
::= { epTrapGroup 1 }


epShutDownTrap NOTIFICATION-TYPE
OBJECTS { masterEPTopolgyMasterRFMacAddress,
masterEPTopolgyEPRFMacAddress}
STATUS current
DESCRIPTION
" EP has gone offline "
::= { epTrapGroup 2 }

masterTrapGroup OBJECT IDENTIFIER ::={hpna31MDUnotification 2 }

masterDiscoveredTrap NOTIFICATION-TYPE
```

```
OBJECTS { ifMasterTopologyInterfaceId,
ifMasterTopologyMasterRFMacAddress,
masterInfoAuth}
STATUS current
DESCRIPTION
" New Master  is discovered "
::= { masterTrapGroup 1 }


masterShutDownTrap NOTIFICATION-TYPE
OBJECTS { ifMasterTopologyInterfaceId,
ifMasterTopologyMasterRFMacAddress}
STATUS current
DESCRIPTION
" Master has gone offline "
::= { masterTrapGroup 2 }

END
```